NASA TECHNICAL NOTE

N73-21880
NASA TN D-7238

NASA TN D-7238

CASE FILE COPY

ADAPTING THE GODDARD RESEARCH AND ENGINEERING MANAGEMENT EXERCISE (GREMEX) TO NONSPACECRAFT ENVIRONMENTS

by Robert O. Wales

Goddard Space Flight Center
Greenbelt, Md. 20771

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1973

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TN D-7238		

4. Title and Subtitle	5. Report Date
Adapting the Goddard Research and Engineering Management Exercise (GREMEX) to Nonspacecraft Environments	April 1973
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Robert O. Wales	G-1080
9. Performing Organization Name and Address	10. Work Unit No.
Goddard Space Flight Center Greenbelt, Maryland 20771	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Note
National Aeronautics and Space Administration Washington, D.C. 20546	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

GREMEX is a computerized training aid for all levels of research and development managers. The computer model used for NASA training simulates development of a spacecraft. Operation of the model is described together with instructions for changing the input-data cards to alter the nomenclature and response of the model for use in other training environments.

17. Key Words (Selected by Author(s))	18. Distribution Statement
Management training Research and development simulation Personnel development Computerized simulation Project management	Unclassified–Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	61	$3.00

*For sale by the National Technical Information Service, Springfield, Virginia 22151.

PREFACE

In early 1962, when aerospace research and development (R&D) projects were approaching peak effort and attention was focused on the technical state of the art and on prestige schedules, the question of effective management of the vast human and financial resources was overshadowed. Recognition of this significant facet of technical research and development, and a conviction that something can be done about it led to what was in itself an R&D effort of considerable magnitude. The result was the conception, design, development, and testing of a sophisticated management exercise in which simulation techniques and the versatility of computers are employed to provide experience in R&D project management.

Cooperation by a number of organizations resulted in development of the Goddard Research and Engineering Management Exercise (GREMEX). The simulation exercise was conceived by Dr. Michael J. Vaccaro, the GSFC Director for Administration and Management, and developed under his direction. The mathematical model was formulated under contract by Management Technology, Inc. The initial operational computer program was contributed by the IBM Data Processing and Federal Systems Divisions. Development, programing, and the conduct of three feasibility demonstrations were planned and successfully executed by Milton F. Denault, Head of the GSFC Management Analysis and Information Systems Branch.

The feasibility demonstrations included participants from Government, industry, and university research communities. Their enthusiastic response indicates that GREMEX has great potential as a means of teaching R&D executives about the inherent problems of project management, their analysis and evaluation, and the means of dealing with them. Thus GREMEX offers a unique opportunity for transmitting management technology by way of a simulated project management environment.

GREMEX was initiated as a regular career-development training exercise for GSFC technical and management personnel in January 1968.

The program listings for the IBM 7094 and 360 systems computers, as well as the associated Orbiting Optical Observatory (OOO) project data, player's manual, instructor's manual, operations manual, etc., are available to the general public. They may be obtained at cost from the Computer Software Management Information Center (COSMIC) established by NASA at the Computer Center, University of Georgia, Athens, Ga. 30601, to disseminate NASA-developed computer programs.

The purpose of this report is to present a general description of the GREMEX program as seen by the student and by the instructor and of its application to various teaching problems. There follows an explanation of the parameters used in the model, details for preparing a new input to the GREMEX program to replace the spacecraft-model data input, and programing details through which a FORTRAN programer may produce variations in the report system to represent the reports normally encountered by the student in an application different from that of a NASA project.

CONTENTS

CONTENTS—*concluded*

Page

ADAPTING THE GODDARD RESEARCH AND ENGINEERING MANAGEMENT EXERCISE (GREMEX) TO NONSPACECRAFT ENVIRONMENTS

Robert O. Wales
Goddard Space Flight Center

GAMES AND SIMULATIONS IN ADMINISTRATION

In the social sciences, games are currently most used in the context of war gaming, political gaming, and management gaming. All three types of games can be used purely as self-contained games involving actors whose strategies are delimited only by a series of rules and not by any environment. All three, however, can also become games that are played in simulated environments for the purpose of lending reality to the game; this is especially true of management games in general, and of GREMEX in particular.

In all these games the pattern is substantially the same. The players are usually divided into small teams charged with managing the affairs of a company or of an industry for a period of years. This is accomplished by making a round of company decisions representing major policies for the next month or quarter. The decisions range over a number of things: choosing the selling prices and mixture of the products (under conditions of competition, oligopoly, and monopoly), deciding how much to spend for marketing activities, determining outlay for research and development (R&D), choosing the optimal conditions for warehousing of inventory, and selecting production runs.

Each set of decisions is fed into a computer (the game environment) that is mathematically programed to simulate the effects that decisions will have on the company for the next financial period. The computer returns a balance sheet and other data (usually specifically requested by the players) pertinent to the company's position at the beginning of a new month or quarter. This feedback informs the team on such matters as whether they have doomed the company to irretrievable penury, whether they were unable to sell their products because of poor warehousing policy, or whether they are the beneficiaries of some unexpected breakthrough in R&D. These data then become the basis for the next series of decisions, and so forth.

Similarities Among Management Games

Whereas management games have great differences, it is possible to abstract three general features that they have in common. First, there is a feedback mechanism. The environment has been simulated on a computer so that it is possible to return to the players data on the outcomes of their decisions. Players then immediately learn the results of previous decisions and are in a position to identify problems and correct past mistakes on the next round.

Second, most management games are premised on the idea of a fruitful interaction of the players on a team as a precondition to success. Because most of the games involve team competitions, ways must be devised by the company "president" and his "vice presidents" to divide the work and produce mutually acceptable solutions to company problems. Also, company decisions must be made in a limited period of time, and cooperation among the players is a prerequisite of team success. Failure to understand social roles is evidenced in the inability to the team to produce viable decisions.

Third, the games are relatively simple. The simplicity results from the fact that social scientists are still learning about the real world and are, perforce, unable to reduce the complete reality of business environments to the mathematical models essential to playing these games. Simplicity is also desirable because the games become playable as a result. The attention of the players is focused not upon all the complexities of real life, but merely upon those problems defined by the game. A great advantage of this simplicity is that the analytical powers of the players can be severely taxed, but quickly evaluated because there are limited problems to be solved. The solutions found by the players can produce results ranging from a hopeless mess of company affairs to evidence of a highly successful enterprise. Although few of the business games have unique solutions founded on simplistic notions of short-run maximization of profits, it is nevertheless possible to talk about "good" and "bad" decisions in terms of the measurable results that each round of decisions produces.

Usefulness of GREMEX

GREMEX incorporates many features of business gaming into a simulated environment that can apply to public bureaucracies, together with several institutional dimensions of R&D administration as they exist in the Federal Government. The presentation of GREMEX may vary, depending upon the participants involved, the number of decisions to be made, and the availability of facilities. Regardless of the form of presentation, the GREMEX environment is a complex simulation (fig. 1) of R&D administration consisting of the three primary elements: computer model, referee-instructor, and participants.

The computer model predicts contractor performance based upon the decisions each team makes. The model contains equations representing typical trends in contractor performance, common effects of various management decisions made in the life of a project, and the impact of unexpected perturbations (e.g., research failure, unforeseen costs, etc.) on project performance. Teams interact with the computer independently of all other teams. For each round of team decisions, the computer produces a series of reports requested by the team and indicating the status of the project at the end of the month for which the decisions were made.

The referee-instructor plays the roles of contractor and NASA personnel to give the environment reality and personality. It is also the referee-instructor who outlines the structure of the game and encourages a free exchange of ideas and information among participants.

The GREMEX participants bring their own skills to the exercise. Players learn from each other as well as from the instructors and the computer model. This learning process is accomplished in part through the interaction of the participants in teams of three or four. Throughout the exercise the players rotate management assignments within their teams; for example: project manager, project coordinator, financial analyst, schedule analyst, and experiment coordinator. Equally significant is

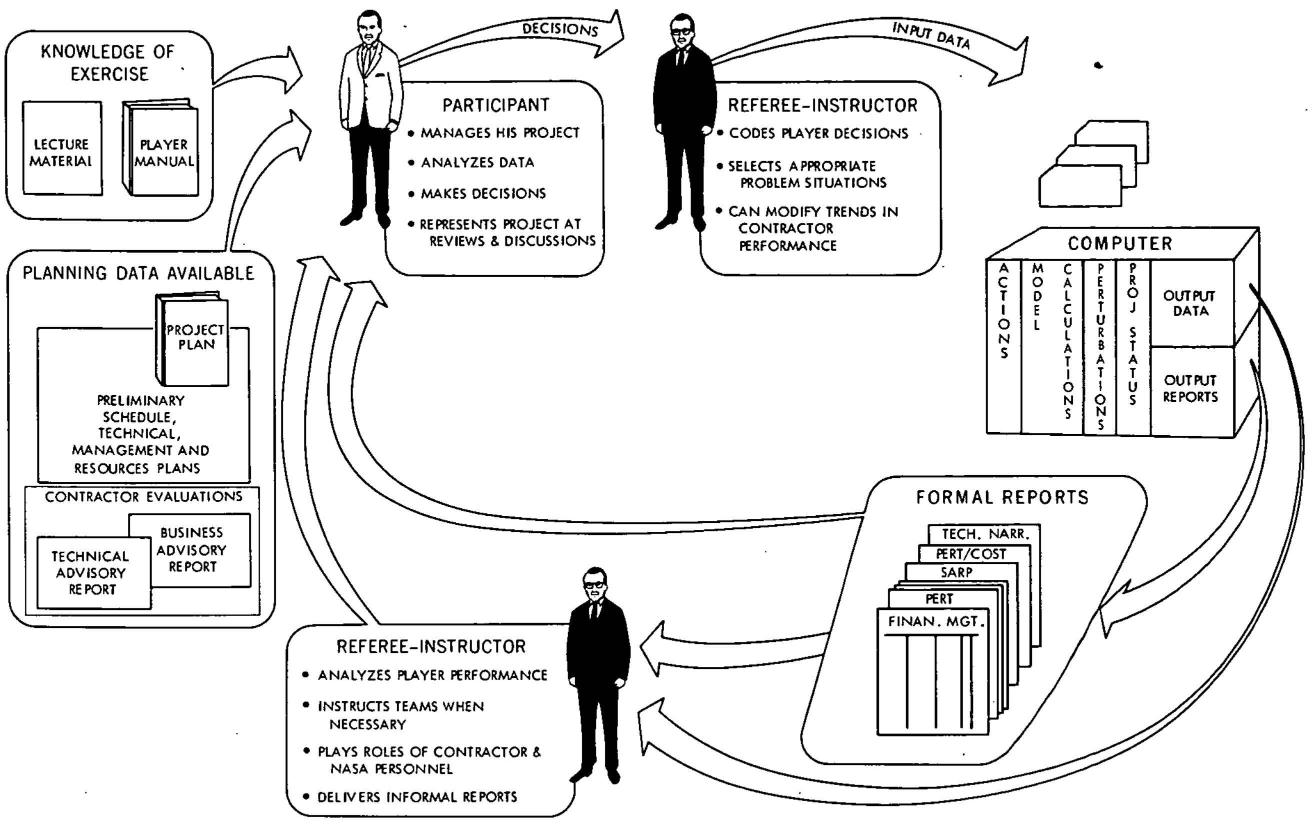


Figure 1.—The GREMEX system.

the interaction that occurs when teams give a status report (oral presentation) to the participants of other teams.

The exercise ends, not when one team has defeated another, but when the learning objectives of the game have been met. The emphasis of GREMEX is to expose participants to many of the factors involved in decisionmaking when managing a project in a Government R&D environment. A management team can win only in the sense that the cost, schedule, and technical performance goals *established by the team and the referee-instructor* have been surpassed.

In this sense, GREMEX poses no "one best way" to manage a project. Some teams clearly surpass others in their use of effective management techniques. Others use the license afforded by the simulated environment to experiment with management methods they could not risk in real life. The serious management experimenters may be the real winners of GREMEX, for they are taking full advantage of the simulation, although their actual achievements in projected cost, schedule, and technical performance may be less than teams taking more conservative approaches to exercise play. It becomes evident that "good" and "bad" decisions, and "good" and "bad" team accomplishments can be measured only in terms relative to the conditions and value assumptions under which the decisions were made.

The assumption of GREMEX is that there are several educational payoffs that accrue to playing the exercise and that these benefits have something to do with training better managers. Empirical research has yet to demonstrate the educational value of management games such as GREMEX. An empirical assessment, however, is currently being undertaken under a grant to the University of Maryland. Not only have responses to questions indicated that GREMEX participants liked the exercise (for undetermined kinds of reasons), responses also showed that participants have found the exercise educationally beneficial. Finally, the respondents strongly indicated that an ability to play GREMEX well was not restricted to scientists, engineers, and trained technicians, in spite of the fact that the project involved scientific R&D.

The organizational setting of GREMEX is that of R&D administration in a Federal agency. At the very least, the exercise familiarizes players with the dimensions of governmental project administration, the relevant roles involved in team management, and the various devices for project control by its manager. At the most, however, GREMEX simulates many features of both hardware and software R&D administration, which has become a major occupation of the U.S. Government.

Furthermore, GREMEX simulates the relationships between complicated subsystems of activity that must be integrated into sequences of achievements contributing to the attainment of a new goal. The obvious examples in Government occur in space and defense weapons systems, but there are other possibilities as well. To the extent that management of the environment, or of a new health system, or of new systems of transportation requires the same kind of directed activity toward goals never before reached, the possibility exists that a simulation such as GREMEX can do much to educate the people who will administer these projects.

The GREMEX computer program is not a self-contained course. It is a "hands-on" training aid that permits the student to see the effects of many types of management actions on a model project. It also provides training in the use of simple management information system reports, showing both the results of insufficient report detail and the problems of excessive detail. This phase of the training is supplemented by an in-basket simulation or other paperwork, including requirements for the project manager to make reports and presentations to higher management.

The computer program is noncompetitive in that each team's data are kept separate and do not react with those of any other team.

Because there is no clear division between "simulations" and "games," these terms will be used interchangeably, as will the terms "player-student-team" and "referee-instructor."

GREMEX IN RELATION TO THE STUDENT

The players are presented with the project and are told that basically there are no ground rules, although standard company policy should be their guide. They are permitted a wide choice of actions. Each player, or team of players, is assigned a referee-instructor. The referee serves as the interface between the players and the computer programs and he converts the player's decisions to a form that the model will react to or, in many cases, he deliberately takes no action to affect the model.

Initially the players may be presented with a choice between a number of contractors who bid to work on the project. Four or more may be qualified, but the lowest bidders all have minor

weaknesses. This paperwork phase of the play may serve as instruction in types of contracts, in reading bidder proposals, or in such similar problems as the instructor may desire. The supporting documentation for the various contractors should be only as accurate as that found in real life. However, it should not lead the player into any surprises in later performance of the contract. That is, if the player selects a contractor whose initial proposals show weakness in the area of cost estimating, then the player would not be surprised if costs in his project rose substantially. Consequently, the game designer must set up rather extensive documentation for the players to use initially and must have corresponding results available in the program through values entered in the preliminary, or PREEP, files as described in later sections. Table 1 is a summary of data as used in the NASA OOO project (a hypothetical Orbiting Optical Observatory).

The choice of a prime contractor only begins a series of decisions that must be made by the project management team. What sort of contract should be negotiated with the private contractor to make him responsive to a Government agency? One possibility is a firm-fixed-price (FFP) contract, which fixes the amount of money a project will cost the Government and so introduces an element of cost predictability in R&D administration. But R&D is risky business, and private companies protect their interests in an FFP by padding their bids to protect themselves against unforeseen contingencies in research and by "shaving" their delivery of services when profits are threatened. A second possibility is a cost-plus-fixed-fee (CPFF) contract, which pays for all allowable costs of R&D and guarantees a fee to cover company profits. This strategy probably enhances the delivery of adequate technical performance (reliability), but provides no incentive for companies to be efficient in their use of resources. Furthermore, CPFF provides a strong motive for companies to be highly cost-optimistic (i.e., seriously underestimating the cost of R&D for the sake of getting the contract). There are various other contract types between the FFP and CPFF that introduce incentive awards for good performance by a contractor because he produces the product on schedule, meets cost and efficiency expectations, or improves upon the expected technical performance. If an incentive contract is chosen, the instructor will require the team to specify the basis for paying the fee and to evaluate the contractor's performance at regular intervals, an additional workload for the team. He will, in general, report that the contractor refused to accept an FFP contract for any but the simplest items if research is required. This is primarily because the project manager has no significant control over such a contractor and thus there is little educational value to the game from that type of contract. Table 2 gives typical program modifiers used for various contract types.

The players must also establish for each contract the type, frequency, and level of detail of the reports desired. This will affect both contract cost and performance.

In addition to the prime contract that represents the major effort on the project, the player must establish a number of secondary contracts for equipment to be furnished to the prime contractor. In the OOO project game, these are the various sets of experiment hardware obtained from university or Government laboratories that will be integrated into the spacecraft before delivery. The student may be presented with an excess of experimenters and must decide whether to expend project funds on backup experiments. In general, various experiments require different lead times, and the player faces the necessity of establishing different contract-award dates for each—based on his evaluation of the schedule, cost, and technical risks implied in each proposal. Because these contracts require the experimenter to retain staff under contract for postlaunch data analysis, the player cannot afford to

Table 1.–OOO Project: Synopsis of Source Evaluation Board (SEB) Findings and Corresponding Model Parameters

Material given student					Model parameters			
Offeror	Technical rank (100 to 500 points)	Business rank (20 to 100 points)	Cost rank	Significant strengths and weaknesses	Probability additives			Overhead multiplier
					Time	Cost	Reliability	
Company A	5 (259)	5 (74)	5 ($3 975 000)	Test facilities among best in United States. Communications and data-handling systems design concept *unacceptable*. Test plans unsatisfactory. Had large cost and schedule overruns on 3 satellite projects.	-0.30	-0.28	-0.45	2.86
Company B	4 (266)	3 (78)	1 ($3 150 000)	System-integration design poor. Considerable satellite experience but had generally poor performance. Has 100 percent cost overrun on current Air Force contract. Cost of doing business (overhead) is low.	-.28	-.30	-.33	2.24
Company C	2 (364)	4 (75)	4 ([a]$3 495 000)	System and subsystem design concepts above average. Test facilities will have to be enlarged. Had 20 percent cost overrun on previous work. Proposal requires $675 000 in GFE from *current contract*.	-.10	-.05	-.02	2.51
Company D	1 (366)	1 (tie) (85)	3 ($3 410 000)	Qualifying experience weak. Spent $800 000 (nonreimbursable) toward this contract. Basic design concept good, especially thermal control, but spacecraft needs more power, experiment space, and data-handling capacity. Management proposal excellent. Has workers on strike. Cost experience good.	0	0	0	2.45
Company E	3 (284)	1 (tie) (85)	2 ($3 170 000)	Good satellite experience. Stabilization and control systems strong, but basic design concept marginal. Reliability below average. Elaborate test facility must be constructed. Management proposal excellent. Has had no major cost overrun on Government contracts. Has lowest cost of doing business.	-.05	-.02	-.10	2.26

[a]Includes $675 000 in Government-furnished equipment (GFE).

Table 2.–Model Parameters for Various Contract Types

Contract type	Probability multiplier, percent		
	Time	Cost	Reliability
Cost plus fixed fee (CPFF)	-15	-15	20
Cost plus incentive fee (CPIF)	-5	-10	25
Cost plus award fee (CPAF)	-5	-10	25
Firm fixed price (FFP)	-10	none	-10
Fixed price incentive (FPI)	-5	10	-5
Cost reimbursable (CR)	-15	-10	-5

deliver the experiment hardware too early. In other areas, the program will incur cost penalties for idle manpower waiting for component delivery.

While these contracts are being awarded, the player is establishing (on paper) a project staff including specialists in the areas where particular contractors seem weak, but at the same time within a preestablished total staff limit.

In the beginning of the program, each team has been given the same total dollar funds. After a few contractor's reports have been received, the team prepares a quarterly (or yearly) spending plan for the project. This plan will return to haunt them when they later direct extra contractor efforts and the referee refuses (as contracting officer) to issue the change if there are insufficient funds available for that fiscal year.

At the option of the instructor, the student sees a time lag in his information system: in the month of August his latest reports are for July but his decisions will not take effect until September. For more current information on specific problems the student may "phone" or "visit" the contractor (instructor).

By the time 6 or 7 months (plays) of the project have elapsed, the students should have a good grasp of the reports and have all of the minor contracts and the subsystems in the main contract properly timed for delivery. Occasional technical problems are being solved and a firm budget has been established. At this point the student receives a request from higher management to predict the earliest possible delivery of his system (with added costs) and the effect on reliability. Because the project was originally planned for a prototype and a final unit, one possibility is to upgrade and deliver a prototype. Other possibilities include appropriate use of overtime or multiple shifts. The student is permitted several planning runs that provide schedule and budget-estimate reports. His final report to management may be based on a combination of these runs or an extrapolation including decisions not considered in one of his runs. Management now takes his report under consideration and he must continue to manage the project for several months without knowing whether or how much to accelerate the completion date.

When the go-ahead for an accelerated delivery is received, it may be for a different date than the student had recommended and the additional funds also may be different.

In addition to preparing written monthly management reports, the player presents a periodic oral review at which the other teams play the part of NASA management and are expected to critique the problems and solutions reported. It should be emphasized that the computer program operates separately for each team and the game is noncompetitive. Although the teams have the same basic goal, the differences in their decisions and the influence of the instructors combine with random elements in the program (as explained later) to provide wide variations in the situation for each team at any given month in the project. Thus, while the teams should see similarities in each other's problems at the critiques, they should not be able to say "we did it this way" for an identical problem.

When the project is about three-fourths complete, most of the minor contracts and subsystems have merged into a single unit and most of the manager's options have been played out. At this point the simulation may stop and the instructors may give a final critique to all teams.

ROLE OF THE EXERCISE REFEREE-INSTRUCTOR

The referee is the key individual in GREMEX. He is responsible for establishing the proper teaching and learning atmosphere, for developing management problem situations, and for the performance of game mechanics.

The referee performs a liaison role. He facilitates the interface between the player and management when players require further information on objectives or represents contractors to explain schedule changes or design problems. The referee is expected to answer all questions concerning the simulated environment. Such questions may demand that the referee invent rationalizations for situations that appear in reports when the causes are not readily apparent in the computer system output.

Although the referee does not interpose his own convictions or recommend solutions, he does attempt to develop player behavior so that the team arrives at a greater understanding of management problems. In particular, the referee reinforces known "good" and "bad" decisions verbally and through taking actions that affect the computer model. The team is permitted to make mistakes, but it is encouraged to see its errors. A spirit of self-criticism is fostered. Should errors be overlooked when examining reports on the results of team actions, the referee may point these out. The referee helps to resolve deadlocks and differences of opinion between team players, when necessary.

The optimum number of players is three per team; however, a referee can adequately handle four. Fewer than three may be allowed, but not more than five, because the greater number tends to reduce participation of each individual. At intervals the team members exchange the roles of project manager, financial analyst, schedule analyst, spacecraft manager, experiment manager, etc. To foster fruitful development of the group decisionmaking process, the referee must closely monitor the actions and results of each month's play. Usually, no more than one team can be supported by a referee working full time.

It is important to maintain team integrity. Each player should participate in each month's decisionmaking conference and should have arrived at a set of tentative recommendations before each conference convenes. The dynamic nature of the learning process demands that not too long a time period elapse between conferences. It is necessary that player interest be stimulated and that the plays

be slow enough to permit in-depth analysis of progress but fast enough to preclude boredom. Therefore, the interval between monthly plays should change as the project advances.

A game of around 16 to 20 plays, lasting through a 5-day workweek, usually is adequate to accomplish the objectives of GREMEX. A standard recommended number of plays is four per day, although during the first day it will probably not be possible to complete more than two or three plays because of the time required for organization, indoctrination, etc.

An initial lecture should be presented by the instructor at least 1 day prior to start of play. At this lecture the students are introduced to the technical and financial details of the project and are given a review of the various types and levels of reports. They are given the background documents (project feasibility studies, experiment proposals, management resource allocations) and the bidders' proposals or the source board evaluations for a number of contractors. Teams are not assigned at this time, but each student is directed to choose a contractor and be ready to defend his choice for the opening play.

Instructors assign players to teams by attempting to balance technical experience and skills if there are varied backgrounds in the group. As the team members defend their individual choices to arrive at a team decision, the referee observes their strengths and weaknesses and begins to manipulate the project to emphasize appropriate types of problems. For example, one of the potential contractors is reported to be on strike. If the team asks about the strike, the referee may report the strike as settled (with or without a wage increase) or may report it as still in progress in a manner to encourage or discourage further consideration of this contractor. The basis for the referee's decision might be whether the other potential deficiencies of this contractor are likely to be useful for a teaching situation later in the game, or he may wish to distract the team from this contractor so that the various teams will not begin with the same contractors.

The referee accepts the team's written decisions and converts them to a GREMEX input. Each possible decision has previously been assigned a code number by the game designer, as described in a later section on the PREEP program. These decisions are called "actions." The referee selects prepunched cards containing these action numbers together with a header card containing the team number and month number. For the first input he may require 2 months' decisions to be able to simulate the normal information time lag by returning only the first month's reports from which the team must make the third month's decisions.

The mechanics of computer job submission will depend on the local computer system. One of the instructors or an additional person will collect the inputs from each referee and assemble them with the necessary job control cards for submission. Some team decisions may require punching additional input cards. Computer printouts should be on multicopy paper to provide a copy for each student and one for the instructor. The first few pages of each are special reports for the instructor and are not normally given to the students. It is also helpful to the instructor to receive the next month's reports as an aid in his role playing. He also should check these as soon as possible for any play errors so that the input may be revised and rerun before the output is due to the team. The team should be protected against program, referee, or computer errors that distract from the realism of the simulation.

Upon the receipt of the first printouts, the instructors usually will go through them with the students in detail. The lesson plan or scenario should allow extra time for this.

The first few months' plays usually are devoted to establishing the prime and secondary contracts and to shifting manpower within the subsystems of the prime contract to establish proper delivery times.

The referee requires the team to establish any parameters on which award or incentive fee payments will be made, and he makes computer inputs to reinforce the contractor's performance on these subsystems and degrades others if the team has missed a weakness indicated in the contractor's bid proposal. The team must also set up their project staff, choosing appropriate management and technical specialists. No computer input results, but the referee considers the types of staff available later when problems are to be investigated by a visit to the contractor.

Little problems give way after a time to severe problems. If and when an adequate measure of progress is attained by the team, the referee may want to introduce severe difficulties that will force the team to respond by changing plans. For example, an experiment contract that is proceeding well may suddenly be hampered because an essential piece of equipment becomes unavailable. This kind of major disruption to the orderly flow of work in GREMEX is called a "perturbation." Perturbations are completely under the control of the referee.

These problems should be well formulated by the instructor staff, with background details rehearsed or preprinted technical memoranda and cost/time proposals available for the various expected team decisions. Perturbations are a useful tool to provide the instructor opportunity for a formal lecture on decisionmaking logic.

Because the GREMEX management game is complex, it is not always possible to predict what casual information may be needed by the player. Therefore, the referee is expected to make this determination ad hoc. Sometimes it may seem suitable to invent a rationalization; at other times, to deny rationalization. Sometimes it may be useful to overwhelm players with rationalizations, so that unraveling the mass of detail in itself provides a management challenge; weeding through the mass and learning to identify only the essential causes can be a valuable experience.

After numerous simulated months of play, most kinds of management learning have been experienced. In particular, the team has assessed the wisdom of its actions by viewing their effects upon project progress. By this time the lead teacher may wish to "wrap up" the game. He will do so either by conducting a critique in which the members of the teams and the referees review honestly the entire history of the project, or by having the teams discuss their approaches and progress collectively. Performance trends are examined, team management objectives are identified, solutions to problems are evaluated, and the pros and cons of alternatives are discussed.

At the final critique, or a few plays before, the team should be informed that the referee has been influencing the results for teaching purposes and that the problems encountered were not always the result of the team's decisions.

By design GREMEX is in no way an employee rating device. It is individually noncompetitive. While the trends of contractor performance reported in GREMEX are representative of the real situation, GREMEX cannot be used as a means of predicting exact contractor performance should a

comparable situation develop in the real world. The player is not expected to learn facts and figures that can be directly applied to any real project. GREMEX is not a predicting device and will not forecast for a player the precise consequence of any action which he might take in the real world. It will not indicate an ideal tradeoff between management objective factors such as cost, time, and performance. GREMEX is sufficiently complex, with various random elements, that, just as in real life, no two games (projects) can be exactly alike. Hence, the results of different teams cannot be exactly comparable. For example, if one game should result in a total project cost, time, and performance poorer than in another game, there may well be mitigating circumstances that may justify the differences. Among these can be chance occurrences, different value scales for management goals, timing factors, lack of information, different introduction of problems by the referee, and disagreements among the players as to the best course of action.

GREMEX COMPUTER PROGRAMS

The GREMEX computer programs are basically a program evaluation and review technique (PERT) reporting system. In the usual PERT program the operator inputs each month the amount of work performed on each activity and the computer does the bookkeeping to determine the expected completion date of the project and whether events will be completed ahead of or behind the needed dates (positive or negative slack). GREMEX automatically assumes that all activities due to be worked in the current month will be worked. In regular PERT programs, the expected durations of future activities remain constant unless a change is submitted by the operator. GREMEX, however, predicts new durations (and costs) each month based on management actions taken by the team and the contractor's abilities. This "reestimating" by the contractor provides life to the model. To prepare for this feature, the project model must specify for each activity three parameters in addition to the usual duration and cost estimates. These three are related to the probability that the time estimate is correct, the probability that the cost estimate is correct, and the "probability of reliability" or probability of technical success. Management actions usually can be expected to change these probabilities. For example, use of overtime or double shifts in R&D work will not only decrease the duration and increase the cost by known proportions but will also increase the likelihood of accidents or mistakes, thus decreasing the probability of reliability. Similarly, as more detailed cost reports are requested, better cost control is to be expected; therefore the probability of cost estimate correctness should be a function of the report system established by the team.

The GREMEX programs consist of

(1) An initialization program (PREEP) which accepts specifications of the project PERT network from a punchcard deck, sorts the network topologically, and outputs to magnetic tape separate copies of the network for each team. Also included are files of predetermined referee actions and constants, such as the number of days in each month.

(2) The MAIN GREMEX program operates on the magnetic tape containing the current status of each team in three steps:

  (a) Reads the team input cards, verifies that they are for the right team and month, inserts team's management actions into proper working files, reads current project status from tape 1

(b) Computes effect of contractor's work for the current month and effect of the team's actions on future schedule and costs, writes updated project network onto tape 2

(c) Using updated project status, prints the reports requested on the online printer (or tape 3), and recycles to step (a) for next team

Computations of the MAIN program for time and cost updating will be described before considering the details of the PREEP network cards.

The program is available from COSMIC in either IBM 360 or 7094 versions. The IBM 360 version is coded in FORTRAN except for two minor routines described later. Because of the larger word size in the IBM 7094, the data files are packed and several additional machine-language subroutines are used. Corresponding routines have identical names and the model performance is identical.

Note: The remainder of this report will refer to the files as used in the IBM 360 version.

Time is carried in the program in tenths of a week from the start date of the project. For simplification, the program does not determine Saturday or Sunday dates or holidays but may show work being started or completed on any date. The player should not be concerned with this and in practice will seldom notice it, especially if the project year is not the actual year. While the computation equates a tenth of a week to 0.7 of a day, the player may think in terms of a 5 day week, or one-tenth equals a half day.

Computation of the start date for a particular activity requires that the activity file be sorted and all prior activities in the flowchart be evaluated for management actions and/or completion before the program reaches the particular activity. Sorting is done only in the PREEP program before the first play and thus no further additions or deletions to the network except the predetermined perturbations can occur during play. (For the one exception, see subroutine ACTNT in app. A.)

In the forward pass through the network, the program considers the uncompleted activities. For each activity, a search is made against the player-input working files to accumulate any probability or percentage changes. A new value for activity duration is computed and, based on the completion dates of the predecessor activities, a new estimated completion date is computed. If this date is earlier than the current play date, then this activity will have its just-completed indicator set. This indicator is used in preparing reports to indicate work done and costs incurred for the current month.

To obtain the current duration, the program considers the last reported duration NACTIV (28, J), any absolute or percentage time changes caused by the current month's management decisions, and a replanning multiplier. (The contents of the activity file NACTIV are listed in app. B.)

The replanning multiplier for activity duration is a function of the history of actions taken against this activity, of the basic file data created by the game designer, and of a random number.

The random number is established by a table in subroutine RANDOM. This routine selects a number, based on the month of the year, that will be the same for all players. The table is graded over a 1-yr period, with larger numbers in the winter months. It is expected that the prime contract is started in January and will produce greatest replanning during the initial stages of the contract and again 1 yr later, approximately when hardware tests might be performed.

Project history is obtained by computing the present time probability P_t, or NACTIV (38, J), and comparing it with the last month's probability NACTIV (25, J). These values are not used directly but as entries into a distribution curve for time or cost. Thus for the curve used in the example (fig. 2), a change from a base probability of 98 to one of 90 has less effect than a change from 88 to 80, and probabilities below 50 produce effects of opposite sign. This value is subtracted from the product of the current probability of reliability of this activity NACTIV (40, J) and a fixed reliability character value associated with the activity NACTIV (10, J). Then, by comparing the result K with the current value of the random number (1 to 99), a multiplier M is obtained as follows:

$K >$ random number	$M = 0$
$K \geq 60$	$M = 0.07$
$35 < K < 60$	$M = 0.33$
$4 < K < 35$	$M = 0.50$
$K < 4$	$M = 1.00$

The value of M is used as a multiplier to the activity time deviation NACTIV (8, J), with the sign from the distribution curve changed to obtain the change for this month. An activity whose time probability has decreased usually will result in a longer estimated time, while one whose time probability has increased will show a decrease in duration (depending on the value of M).

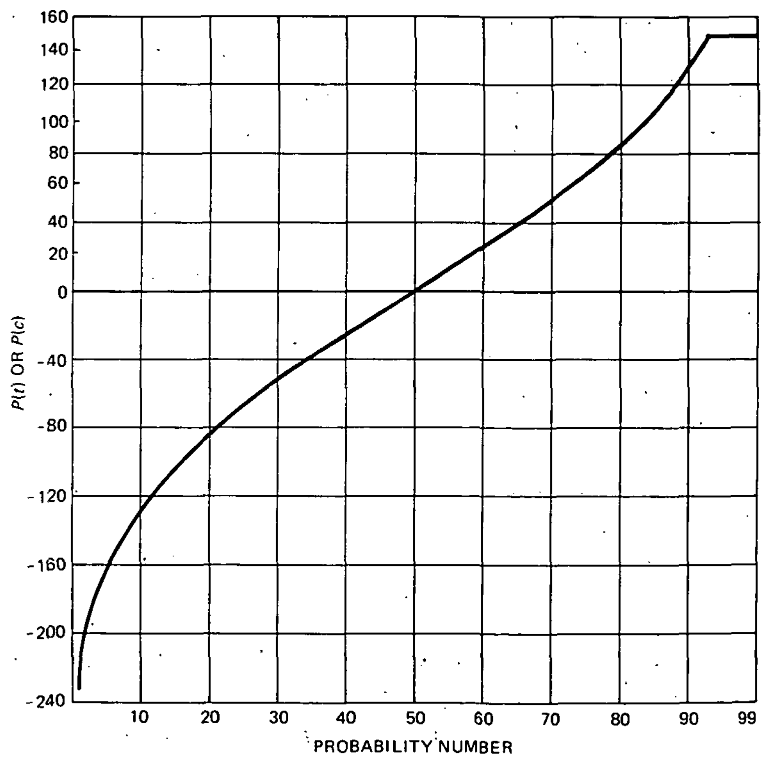


Figure 2.—Example of distribution curve.

Thus the game designer influences the time reprograming in three ways:

(1) K can be made greater than the random number factor more often (i.e., M will be 0 more often) if both NACTIV (10, J) and the initial reliability NACTIV (14, J) are large for this activity.

(2) The historical effect can be changed by adjusting the distribution file—this change affects the whole project, as would a change in the RANDOM routine.

(3) The multiplier NACTIV (8, J) may be set larger or smaller depending on the type of activity. It is typically one-third of the initially planned activity duration.

The reliability probability is used in connection with time changes and, if the activity is designated as a test activity, a comparison of the reliability and the random number status determines whether the test will fail. Failure will double the activity time, as the test will always pass when repeated. On completion of any activity, its reliability value is increased, thus causing a general rise in overall project reliability as work progresses. Project or subsystem reliability is computed as the simple average of all current values for activity probabilities of reliability.

The forward-pass computation is continued for all uncompleted activities. A backward pass is then made to compute the required completion dates and the slack based on meeting a set, scheduled, project completion date (or, if no scheduled date is set by the team, the earliest possible project completion date).

The replanning time changes do not change the activity costs.

New cost estimates are computed in a manner similar to the new time estimates, except that the random number generator and reliability probabilities are not used. Cost probability changes due to player actions for this month are used to obtain values from the distribution curve to multiply the cost deviation factor NACTIV (11, J) for the activity, and free slack charges, if any, are added. Free slack occurs where two activity paths having different slacks come together. For example, in figure 4 (presented later), activity leading from event 8 to event 5 will be completed 22.6 weeks before activity leading from event 4 to event 5. Free slack charges assume that 25 percent of the engineering staff assigned to the early activity must wait for its completion. All costs are computed in thousands of dollars and do not include overhead rates at this stage of the program. Thus if the basic engineering charge for activity from events 8 to 5 was specified as \$3500 (i.e., \$1000 per week), the waiting charge will be $22.6 \times 1000 \times 0.25 = \$5650$. Cost changes are distributed among the engineering, material, subcontract, and technical labor categories in the same proportion as the original costs.

Current values for all parameters are retained in the activity file and summed to obtain new values for the contract and component files. Then all player files are recorded on a save tape for the next play. These values also remain in core storage for the report phase of the program.

Management or referee decisions are initiated by data-card inputs called "*actions.*" The action is usually a single numerical input that changes the probabilities in the system and/or calls some special subroutine to be used for this play only. Different sets of probabilities may be used with the same subroutine by assigning different action numbers. The assignment is made by a data-card file established at network generation time in the PREEP program. The subroutines, however, are contained in the MAIN program and the PREEP file references them by numbers (NCODE). Additional

numbers and subroutines may be easily added in the MAIN program (in subroutine ACTION) or unused subroutines can be removed when computer core storage is critical.

Most actions may be taken either against a single activity or against all activities within a contract, system, or subsystem. There are a few exceptions that apply only to the whole project, such as setting project budgets. If the ACTION file contains entries pertaining to a change in the probabilities or actual values of time, cost, and reliability, these values or changes will be applied either to all activities in the specified contract system or subsystem or to a single activity as specified on the action card by the referee. Changes to these parameters may be expressed either as percentage changes or as absolute changes through different columns on the input card or different action numbers. File entries for P values will be used as percentage multipliers on the activity probabilities if the action number is called. If more than one P multiplier is called for the same activity at the same play period, the average value of the multipliers will be used. Blank entries are not counted in obtaining averages. Entries for S (additive) values are algebraically added to the activity probabilities. Limit checks prevent any probability becoming negative or greater than 99. Other entries provide percentage or absolute changes in present activity duration or cost. Thus in the sample in table 1, the probability differences representing the different capabilities of five contractors are established by five different NCODE 1 actions, and these probabilities will be further varied by probability changes associated with the type of contract and the level of reporting system imposed by the team. Other entries establish the overhead multiplier, specified when awarding contracts, to vary the total contract value, or are used by some subroutines to specify a particular contract or a report type. The referee's input data may, in these cases, require an identical number as a check against the correctness of the action. A final entry provides the basic code (NCODE) to establish the type of action. Appendix A lists the NCODE's and subroutines used together with the data required in the PREEP files and at play input for each.

SPECIFYING THE PROJECT—"PREEP" PROGRAM

In the basic operation of the GREMEX program, the player's network is read from the magnetic tape, updated, and written out on a second magnetic tape. For the initial play, a separate program, PREEP, is provided to establish the initial status of all possible networks for each player. The PREEP program reads initial descriptors of the network from punched cards and prepares duplicate copies of the network for each player (team). The PREEP program is furnished in FORTRAN with dimensions for a large-scale network of up to 999 activities. This program requires a core region of 382 000 bytes for compilation and execution and has a 1-min execution time on an IBM 360/91. The core requirement may be substantially reduced for smaller networks by changing certain dimension statements, as described in the final section, because each activity requires 200 bytes of core in this program.

The first step in establishing the components and the activities is to prepare the master PERT flow diagram. This diagram may be prepared either in event (node) notation or in activity notation. The PREEP program requires numbers to be assigned for both the event boxes and the activity arrows. Later in the MAIN program, input data may reference either the predecessor-successor event number pairs or the activity numbers, and some reports are sorted on the successor numbers. Whereas the event boxes on a flowchart contain descriptive titles related to start or completion of an activity, the descriptors carried in the program are the activity descriptors and imply the passage of time.

For purposes of illustration, a small project, Project Invention, is shown in figures 3 and 4, and sample reports based on this project will be discussed in the final section. In actual teaching situations, the network would be more complex and the activities specified in more detail with additional contracts or subcontracts to provide additional problem areas.

The overall system may be divided, if desired, into several concurrent contracts (fig. 3). The total number of contracts permitted in the system is 15. Each contract may be further divided into systems, subsystems, or components for a maximum of 50 project components. It does not matter whether this division is in terms of separate products or of various segments of a factory, such as engineering or production divisions. The items in the subdivisions need not be contiguous unless there is a possibility that a "cancel work" action might be taken. (See notes on subroutine ACTNT in app. A.)

The master activity chart at this time should also include all perturbations. The perturbation loops may not contain identical activity numbers, but may contain event numbers identical to those in the normal flow of the chart. Confusion during play can result from identical event pairs because actions (overtime, for example) taken against the event pair in the main chart are not transferred to the pair in the perturbation when it is activated.

The PREEP program checks for a limited number of errors in the data cards, such as cards being out of sequence or input data punched in the wrong columns. It does not check the logic of the flowchart for loops or loose ends. If an error is discovered, the program terminates with an error message as shown in appendix C.

All entries are right justified in the data fields. Remarks fields are provided on most cards, but these are not read by the program.

The first card read by the program must be a "type 75" card containing the number 75 in columns 1 and 2 and the total number of teams in columns 3 and 4. The number read in for the number of players (or teams) will later produce that number of identical files on the magnetic tape. The remainder of the groups of data cards may be read in any order with one exception. The description of the components, type 20, must be read before the description of the activities, type 10.

Each segment other than the 75 card is preceded by a card with 70 in columns 1 and 2 and the type number in columns 3 and 4. The terminator for each input segment is a card with a 77 in columns 1 and 2.

Each segment is processed after the 77 card is read, and printed output may be produced. These reports require no further comment except for the activity and hardware report discussed under activity cards. A card with 79 in columns 1 and 2 is used to terminate input data. The program then produces the multiple file copies on tape and exits when done.

As supplied from COSMIC, the listing includes a complete set of data cards for the OOO project, which are referenced in the documentation (referee manual, etc.) included. The following sections discuss those portions that must be changed for other types of projects and give samples based on Project Invention (figs. 3 and 4).

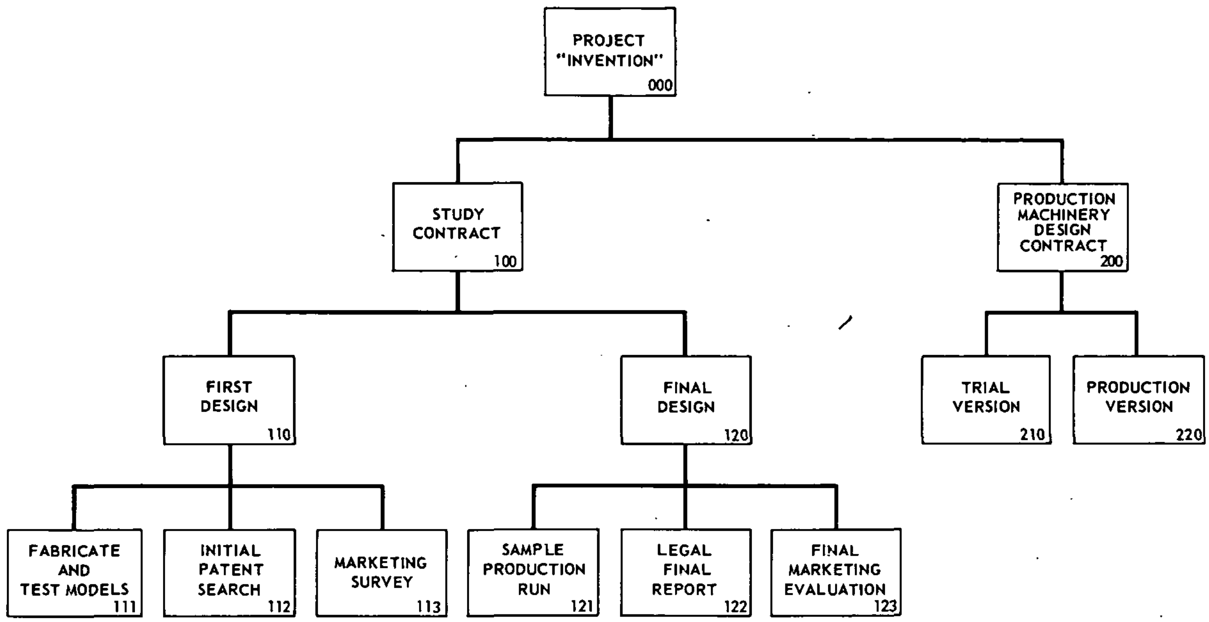


Figure 3.—Work-breakdown structure for Project Invention.

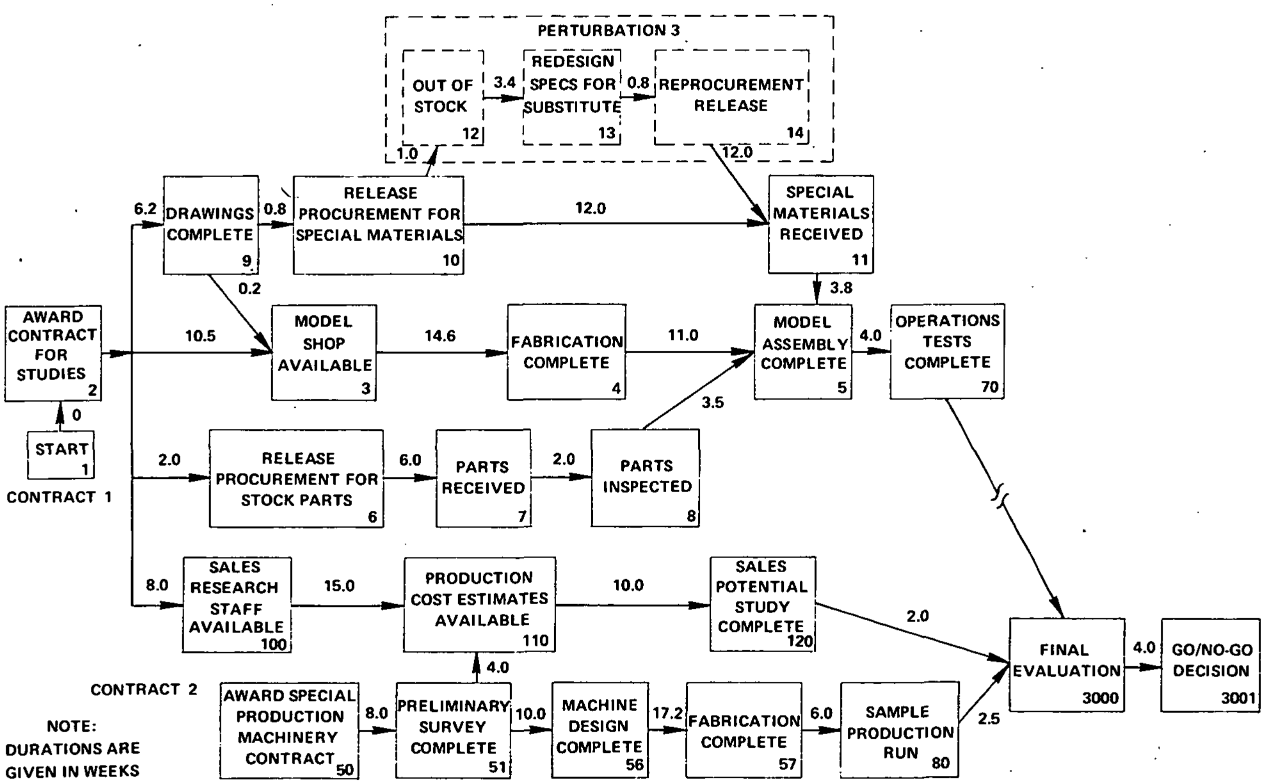


Figure 4.—Activity network for Project Invention.

TYPE		CONTRACT		SYSTEM	SUBSYSTEM													COMPONENT NAME												START DATE (tenths of weeks)									
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
7	0	2	0																	(	H	E	A	D	E	R		—		W	B	S		F	I	L	E	)	
2	0		1	0	0													S	T	U	D	Y		C	N	T	R	C	T										
2	0		1	1	0													F	I	R	S	T		D	E	S	I	G	N										
2	0		1	1	1														M	O	D	E	L		F	A	B	.											
2	0		1	1	2														B	A	S	I	C		L	E	G	A	L										
2	0		1	1	3													M	A	R	K	E	T	I	N	G		S	T										
2	0		1	2	0													F	I	N	A	L		D	E	S	I	G	N										
2	0		1	2	1														P	R	O	D	U	C	T	I	O	N											
2	0		1	2	2														F	I	N	A	L		L	E	G	A	L										
2	0		1	2	3													F	I	N	A	L		M	A	R	K	E	T										
2	0		2	0	0													M	A	C	H	I	N	E	R	Y		C	O								1	5	0
2	0		2	1	0														T	R	I	A	L		V	E	R	S											
2	0		2	2	0														P	R	O	D	U	C	T	I	O	N											
7	7																					(	F	I	L	E		T	E	R	M	I	N	A	T	O	R	)	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40

Figure 5.—Work-breakdown-structure coding form.

Work-Breakdown-Structure Cards

Titles for the work-breakdown structure are entered into PREEP via the type 20 component file. Entries are coded as shown in figure 5. These titles will be used for player reports. Start-time entries may be made for contracts other than contract 1 to minimize free slack at this time. Entries are made in weeks from time of start of contract 1. The values used will be stored and printed as base dates in some reports.

Activity Cards

Associated with each activity must be the following data punched in the card as shown in figure 6. The card type 10 identifies activity cards. Activity numbers must be sorted for input in ascending order; there may be missing numbers, but the upper limit is 999. The predecessor and successor event numbers (or node numbers) may be four-digit numbers in any sequence. Also on the card will be the contract number and the system and the subsystem to which this activity applies, which is used for sorting on reports.

If the activity is to be added by a perturbation, columns 18 and 19 contain the perturbation number. Only added activities may have the perturbation number. This number must be consistent with those contained in the perturbation file (type 30 data). No program checks for this consistency

TYPE	ACTIVITY NUMBER	PREDECESSOR EVENT	SUCCESSOR EVENT	CONTRACT	SYSTEM	SUBSYSTEM	PERTURBATION	$P(t)$	$P(c)$	$P(r)$	DURATION	TIME DEVIATION	INITIAL EVENT	LEVEL	RLC	(blank)	END EVENT	TEST EVENT	COMPONENT END	TOTAL K$	ENGINEERING K$	MATERIALS K$	TECHNICAL K$	SUBCONTRACT K$	COST DEVIATION	DESCRIPTION OF ACTIVITY
70	10																									(FILE HEADER CARD)
10	1	1	2	1	1	1					0		1													AWARD PRIME CONTRACT
10	2	2	9	1	1	1		70	80	85	62	30		3						15	6	1	8		5	PREPARE SHOP DRAWINGS
10	4	9	10	1	1	1		95	95	95	8	1		3						1	1				1	PREPARE SPECIAL SPECS
10	5	10	12	1	1	1	3	90	90	99	10	0														OUT OF STOCK NOTICE RCVD
10	6	12	13	1	1	1	3	90	90	99	34	0								5	4		1			SUBSTITUTE DESIGN COMPL.
10	7	13	14	1	1	1	3	90	90	99	8	0								1	1					FIND NEW SOURCE / ORDER
10	8	14	11	1	1	1	3	80	80	80	160	40		3	60					65				65	10	SPECIAL MATERIALS RCVD.
10	9	50	51	2	1	0		80	70	90	120	10	1	3	90					10	10				2	SURVEY PRODUCTION METHOD
10	10	10	11	1	1	1		80	95	90	120	40		3	70					50				50	10	SPECIAL MATERIALS RCVD.
10	11	11	5	1	1	1		70	80	80	38	15		2	80					9	1	3	5		5	ASSEMBLE SPECIAL PARTS
10	12	5	70	1	1	1		75	90	60	40	40		1	35			1		7	4		3		1	DURABILITY TESTS
10	13	4	5	1	1	1		70	70	85	110	35		1	80					42	10	7	25	0	20	ASSEMBLE BASIC FRAMES
10	15	51	110	2	1	0		90	90	90	40	10		2						1	1				0	EVALUATE SURVEY REPORT
10	17	80	3000	1	2	1		90	90	90	25	25		2	60				1	3	3					APPROVE MACHINE FUNCTION
10	19	3000	3001	1	2	3		80	99	99	25	25		1	50		1			5	5				0	STAFF FINAL REVIEW
77																										(FILE TERMINATOR)

Figure 6.—Activity coding form.

are made until execution of the MAIN program. (See app. D.) An activity with a perturbation number starts as a nonexistent item (similar to a system for which the contract has not been awarded). Management or referee actions against this activity are not permitted until the perturbation is effective; if actions are attempted, error messages will be generated.

Columns 20 through 25 contain the basic probabilities of the activity's operation. Columns 20 and 21 contain the time-probability factor. This is the probability that the time estimate for completion in weeks is essentially correct. If the number is in the region of 60 to 80, the time probability will change substantially during the duration of the game. If the number is in the region of 90 to 99, there will be small changes in the time probability during the game. The cost-probability number represents potential additional changes in the contractor's estimated cost. The reliability-probability number is used in determining the replanning and, if the activity is a test activity, the reliability-probability number together with the reliability-character number determine whether the test will fail. (Methods of time and cost replanning were discussed in the previous section, entitled "GREMEX Computer Programs.") Perturbation activities usually have high-probability entries, because little time would elapse in real life between preparation of the contractor's estimate and performance of the work.

Columns 26 through 28 contain the initial planned duration of the activity in tenths of weeks (no decimal point permitted).

Columns 29 through 31 contain the time-deviation multiplier in tenths of weeks. Generally speaking, this should be about one-third of the original duration for activities other than tests, and less than one-third for activities that would show minimum changes. For test activities, it should be the amount of time required if the test fails. That is, it usually will be equal to the original test time. The activity time will also be reset to this value by an error routine if any action creates a negative duration.

Column 32 contains a 1 for the first activity on any particular contract. Only one activity should be designated to start a contract. See the use of events 1 and 2 in figure 4 for an example where four activities start after contract award.

Column 33 contains a level number used in reporting. It may be a 1, 2, or 3, or no entry for level 0.

Columns 34 and 35 are the reliability-character number used in connection with the random number generator and test failures as previously explained. A reliability character of 90 or above has a low probability of failure. A reliability character of 70 or below has a high probability of test failure. A blank entry will be converted later to a value of 100.

Column 37 contains a 1 for the single activity that is the last in the overall network.

Column 38 contains a 1 for activities that are to be considered test activities. If a perturbation series is to be used in connection with a test failure, an activity (described as "test failure" with 0 weeks duration) having this indicator set to 1 will print on the technical narrative report. (See fig. E-5.)

Column 39 contains a 1 for an activity that is the terminal activity of a component. This is an activity all of whose successor events are in another component.

Columns 40 and 41 contain total cost (in thousands of dollars) initially associated with this activity. This cost must agree with the sum of the next four entries. Cost values will be multiplied in the main program by the overhead multiplier established by an "award contract" action.

Columns 42 and 43 contain the engineering-labor charge. Note that one-fourth of the engineering-labor rate is also used for free slack charges and may be set equal to zero if free slack charges are not appropriate.

Columns 44 and 45 are the material costs, columns 46 and 47 are technical-labor costs, and columns 48 and 49 are subcontract costs.

The distinctions between these four labor costs are used primarily for reporting. It is not possible to change only a single cost during the progress of the program, but all four costs may be changed by the same percentage.

Columns 50 and 51 contain the cost deviation used in contractor cost reprograming. Generally speaking, this would be on the order of one-third of the total cost of the activity, except for a test activity, where it should be on the order of the original total cost. This is also a reset value used if costs become negative through referee error.

Columns 52 through 75 contain the literal description of the work of the activity that will be printed out in all reports.

The final card in the activity deck must be a type 77 card containing no other entries.

As mentioned previously, the type 10 cards must follow the type 20 cards. The remainder of the files may be put in any order.

At the completion of the type 10 card input processing, a hardware report or component summary and an activity listing are printed.

The hardware report lists the costs of each system and subsystem by summing all activity cards for each subsystem. These costs are before overhead. A completion time in weeks is also computed and printed for each component level. Contract 1 is assumed to start at time zero. The other contracts may have best estimate start dates introduced in their type 20 cards; otherwise they are also assumed to start at time zero.

The activity report lists the input data for each activity, as well as expected completion times and slack values for each activity except for perturbation activities. Where two or more activities terminate in the same successor event, free slack is listed for the early completion of activities. It is assumed that the specified costs for these activities already include these free slack charges, and no changes in costs are computed. The best start times for the other contracts may be found and inserted into the type 20 cards to produce minimum free slack times in the base data files. The activity report is useful to the referee's role playing, in reporting the proportion of an activity that may be material or subcontract costs.

Perturbation File

The perturbation file is made up from card types 30, 31, and 32. These three cards must appear in succession for each perturbation. (An example of perturbation coding is shown in fig. 7.) All cards contain the perturbation number in columns 3 and 4. This number must match the numbers in columns 18 and 19 of the activity file and is the number used by the referee on the input form. The perturbation card inputs in this file must be sorted in ascending order of perturbation numbers but need not be consecutive. Also on the type 30 card are the numbers of the contract (columns 5 and 6), system (column 7), and subsystem (column 8) in which the perturbation occurs and the activity number for the critical activity (columns 9, 10, and 11). If the critical activity entry is zero (or blank), the perturbation will take effect immediately upon the referee's submission of the perturbation number. If the number is not zero, it indicates the activity that will cause the perturbation to take effect when this activity begins to work in the play of the game if the perturbation number has been submitted. The activities listed on the next two perturbation cards are then respectively deleted and added to the overall network. All activity number fields are three columns long and the first blank group terminates the scan of this card.

The type 31 card contains the activities that can be deleted by the perturbation. Any number up to five activities may appear on this card.

The type 32 card contains the list of up to 10 activities to be added by the perturbation. These are the activities in the activity file containing the same perturbation number as on this card.

There is an overall limit of 90 perturbations for the perturbation file. Upon detecting a terminator type 77 card, the PREEP program prints a listing by activity number of the perturbation activities for the referee's use.

Action File

The action file establishes reference numbers (action numbers) for combinations of model parameter changes and/or special subroutine calls used by the referee.

Submitting a single action number then makes available the various parameter changes and calls the correct logic subroutine (via NCODE) for the desired effect.

Figure 8 shows some of the entries used for the demonstration program; the OOO project uses 78 actions, which are described in the referee manual. Action numbers must be consecutive and are limited to 99.

The major parameters available to the file are the percent change in cost $U(c)$, percent change in time $U(t)$, percent change in probability of time $P(t)$ (or cost, or reliability), and additive change in probability of time $S(t)$ (or cost, or reliability). The P-type entries are based at 50 = 0 percent (i.e., 45 = -5 percent) with blank or zero entries ignored in all cases.

Also used only when awarding contracts is an additional overhead cost multiplier (in percent). This value adjusts the basic contract cost for various bidders. (See table 1.) The value should allow for an additional nominal 10 percent cost for the typical selection of reports. These costs are determined by entries in the $U(c)$ field of the report actions.

Cols 1–2	Cols 3–5	Cols 6–8	Cols 9–11	Cols 12–14	Cols 15–17	REMARKS (cols 40–80)
70	30					START PERTURBATION FILE
30	3	111	10			P#3 SYSTEM/CRITICAL ACTIVITY
31	3	10				P#3 DELETE ACTIVITY #10
32	3	5	6	7	14	P#3 ADDED ACTIVITIES (FOUR)
77						FILE TERMINATOR

Figure 7.–Coding for perturbation.

FILE TYPE (40)	ACTION NUMBER	TIME-PROBABILITY MULTIPLIER $P(t)$	COST-PROBABILITY MULTIPLIER $P(c)$	RELIABILITY-PROBABILITY MULTIPLIER $P(r)$	(NOT USED) (FORMERLY DELAYED PROBABILITY CHANGES)	TIME-PROBABILITY ADDITIVE $S(t)$	COST-PROBABILITY ADDITIVE $S(c)$	RELIABILITY-PROBABILITY ADDITIVE $S(r)$	(NOT USED)	PERCENT TIME CHANGE $U(t)$	PERCENT COST CHANGE $U(c)$	COST DOLLARS CHANGE $G(c)$	OVERHEAD	C/R/L	(NOT USED)	NCODE	REMARKS (NOT USED BY PROGRAM)
70	40																ACTION FILE
40	1					10	10	-5					265	1		1	PRIME TO "A"
40	2						-3						208	1		1	PRIME TO "B"
40	3					-15	5	5					211	1		1	PRIME TO "C"
40	4												180	2		1	MACH. TO "D"
40	5					-5		20					160	2		1	MACH. TO "E"
40	6															2	SET START DATE
40	7															8	SET COMPL DATE
40	8	45	45													14	FIXED FEE CONTR
40	9															14	INCENT FEE ''
40	10		70								2			1		4	TYPE I REPORT
40	11	55	65								1			2		4	'' 2 ''
40	12	60												3		4	'' 3 ''
40	13	70									4			4		4	'' 4 ''
40	14			65										5		4	'' 5 ''
40	15															6	MONTHLY REPORTS
40	16	45	45	45										2		5	LEVEL 2 ''
40	17										1			3		5	LEVEL 3 ''
40	18							-5		-20	20					19	OVERTIME WORK
. .	. .																. . . .
77																	(FILE TERMINATOR)

Figure 8.–Action coding form.

The miscellaneous parameter *C/R/L* in columns 59 and 60 has a variety of uses depending on the particular NCODE. This usually requires an identical entry by the player and serves as a check on proper selection of action.

Appendix A lists the required and optional entries for the 16 subroutines and several blank NCODE's used only for parameter changes.

Calendar File

Card type 50 creates a calendar file consisting of the first three letters of each month of the year and the number of days in each month. No provision is made for leap year or designating which days are Sundays. No check is made for work occurring on holidays and weekends, and generally the players will be too busy to note that something is started or finished on a Sunday. The first card has 50 in columns 1 and 2 followed by the three letters of each month (all 12 months on the same card without blanks). The second card has 50 followed by the number of days in each month (all 12, two digits, no blanks). The program will prepare an additional file entry for identifying each month in terms of tenths of weeks from January 1.

Distribution File

The type 60 file contains the distribution file. These are the 99 numbers that are used in connection with the automatic reprograming as previously described. Each card has 60, sign, and three digits. It is recommended that the distribution in the example of figure 2 (supplied from COSMIC) be used until the program designer has become more familiar with the effects and reactions of the program.

Common File

As the PREEP program has been reading these data cards, it has been determining the total number of entries present and storing these numbers in the COMMON region. When the program detects that all data cards have been read in (i.e., it reads a 79 card), it then repacks the activity file in topological order and writes out copies of all files on the binary tape. The tape will contain the activity file; perturbation file; component file; contract file; action file; distribution file; COMMON file; report-request file (blank at this time); the numerals, months, and days in the year; and a file for referee's information (also blank at this time). Details of the major file contents are given in the appendixes.

Each of these files will be of the size indicated by the program DIMENSION statements regardless of the number of entries. The DIMENSION statements may be reduced to conserve core storage but must be identical in both the PREEP and MAIN programs.

The program will make as many identical copies on the tape (except for the change in player number) as requested by the 75 card.

Error Detection

A limited number of checks are made on the input cards, primarily for proper sequence. Error messages of the form ERROR XXX will be printed and the program will exit without attempt at recovery. An error listing for PREEP is given in appendix C.

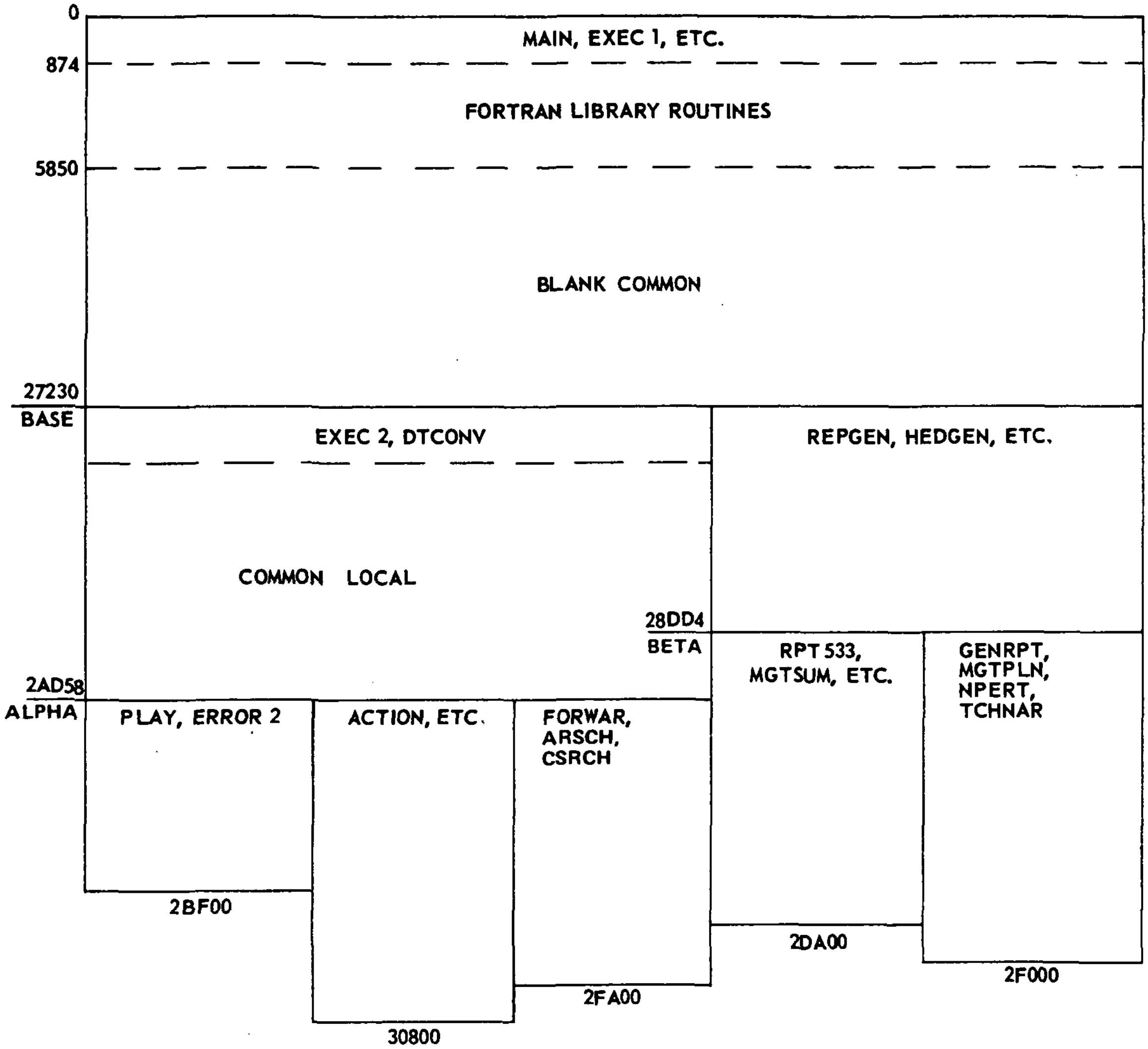
OPERATING DETAILS OF "MAIN" GREMEX PROGRAM

The main GREMEX program consists of about 47 subroutines and a three-statement main-line program. The program divides logically into four major segments that can be loaded as link or overlay programs. When used with small computers, some of the four segments may be further subdivided. The total length of the program, if not overlaid, is 316 000 bytes of core. Run time is approximately ½ min per player on an IBM 360/91. Figure 9 shows a typical overlay example for 246 000 bytes of core. The description of the contents of COMMON and the main arrays in it are listed in appendix B for the IBM 360 version. The IBM 7094 version uses identical input cards and produces identical computations and reports.

The IBM 360 program is written in FORTRAN except for subroutines PACK and UNPAK. These two routines set and read indicators in word 54 of array NACTIV, for which a 16-bit word is expected. These routines, written in IBM 360 assembly language, may be replaced with FORTRAN routines, but program execution will be slower. The approach may be either as a problem in binary (integer) numbers or by bit masking and sensing, depending on the special FORTRAN features available. Subroutine PACK(N, m, J) sets bit m in NACTIV (54, J) to the value of N(0 or 1). Subroutine UNPAK (N, m, J) sets N negative if bit m in NACTIV (54, J) is set to 1, and to 0 if $m = 0$. The value of bit m is not changed.

The MAIN program initializes certain counters and calls the routine EXEC1. Subroutine EXEC1 is the main controlling routine for the program. It continues to cycle through the four parts of the program. The first part of the program reads the team control cards. The second part of the program reads the team action cards and stores appropriate actions. The third part of the program does the mathematical computations to update the PERT network data. The fourth part of the program is the report-generating program. For the example of figure 9, EXEC1 has been subdivided. EXEC2 can be part of EXEC1 if fewer overlays are used.

The controlling routine for the first part of the program is PLAY. This routine obtains the past history of the particular team from magnetic tape (unit 10), stores it into COMMON, then checks the first control card for the player to be sure that it is for the proper player and period. If it does not match, it copies the history data onto the output tape (unit 11) without change and proceeds through the input tape data until the match for the right team number is found. The team numbers must be in ascending sequence. The routine PLAY also checks for final-deck terminators to see whether the whole sequence is finished or whether output data should be copied back onto the input tape for a second set of plays. It is possible, by using the proper terminator, to update several months on one computer submission. When a match between the team number and file data is found, routine PLAY returns to EXEC1, which then calls routine ACTION. Subroutine ACTION together with its various subroutines, ACTNA through ACTNY, process the input data cards, the secondary subroutines being called by the various NCODE's, as explained previously.



OVERLAY BASE
INSERT EXEC 2, DTCONV, LOCAL
OVERLAY ALPHA
INSERT ACTION, SET PTB, RANDOM, ACTNA THROUGH ACTNJ, ACTNN, ACTNS, ACTNT, ACTNU, ACTNX, ACTNY
OVERLAY ALPHA
INSERT FORWAR, ASRCH, CSRCH
OVERLAY BASE
INSERT REPGEN, BCDOUT, HEDGEN, CLNDR, BTOD, CNSTNT, OUTPUT, MISC, CDATE, FETCHN
OVERLAY BETA
INSERT RPT533, MGTSUM, CALCF, COUT 1, COUT 2, CBOUT 1, CBOUT 2, SHIFTT, MSHFT, TSHFT, PLN533
OVERLAY BETA
INSERT MGTPLN, NPERT, TCHNAR, GENRPT

Figure 9.—Overlay core map for IBM 360.

When a team terminator card (a number 1 in column 1) is read, subroutine ACTION returns to the controller EXEC1, which then calls subroutine FORWAR unless error conditions exist. Subroutine FORWAR updates all of the activities and the component files by advancing time by 1 month. During the processing by subroutine FORWAR, some of the referee's reports are generated. This is information such as major changes in activity time and status of the various perturbations in the system. This report is normally not given to the players.

FORWAR itself consists of a forward computation through the network to determine the expected dates and replanning effects and then a backward computation based on the schedule date to determine the required dates and the slack. For a very small computer, FORWAR may be divided into two separate passes, forward and backward. This is the largest single routine in the program and the data are all in COMMON.

On completion of the computations, the updated team history is written out by FORWAR on the output magnetic storage tape (unit 11). FORWAR then returns to EXEC1, which calls the report-generator controller REPGEN.

This routine scans the report-request storage array and then calls various report-generator subroutines as required. On most machines it is possible to load the entire set of report-generating subroutines in one overlay. However, the example in figure 9 shows one method of subdivision of this report section. The report-generator routines prepare team output on unit 6. Because it is usually desirable to give a copy of the output to each team member, one must either load multipart paper on the main-line printer for unit 6, or specify unit 6 as a tape, which is then printed offline. The proper control cards depend on the installation using this system.

The report-generator subroutines do no computation to the basic status of the program, although in some cases they accumulate intermediate sums. These sums are not passed on in the output storage.

At the conclusion of printing all reports, control is returned to EXEC1, which then checks to see if the total number of players specified has been processed. If not, it returns to routine PLAY to look for data cards for the next team to be played.

Normal termination of the program is through routine PLAY reaching the end of the data cards, a deck terminator 2, or on reaching the total number of player files contained on the magnetic tape.

A player's data may be updated more than 1 month in a single computer submission through the use of a terminator 3. This terminator causes all teams on the output tape to be copied back onto the input tape and then returns to EXEC1 for more player input data. The original input team history is destroyed by this action.

Abnormal termination may occur through subroutine ERR, which may be called by ACTION or by FORWAR. If called by ACTION, it is usually a case of incorrect data cards. If called by FORWAR, it may represent bad magnetic tape data being read into COMMON. The terminating routine ERR would cause the team input history data to be copied directly to the output tape, essentially canceling that play for that team. Appendix D lists the errors that may be detected. If there are more player numbers, the terminating routine ERR returns to EXEC1 and play continues with the next player in a normal manner.

Because the FORTRAN input/output checking routines in most computer systems will abort the program in the case of mispunched data cards (letters in a numeric field, etc.), the status of the player's history tape may then be indeterminate. It is recommended that the referee-instructor be supplied with prepunched player input cards for the most common player requests and that the player output tapes from previous plays be saved together with the player input decks until the current inputs have been checked.

During play and computations some other error checks are made. Incorrect action numbers, event pairs, or too many inputs (120 per team) will cause the card to be bypassed and a message printed. If computations cause an activity time or cost to go negative, it will be reset to the DEVIATION value without a message. The upper limit for activity duration or slack is 5 yr (260 weeks) and for a single activity cost is $100 000. The game designer may wish to convert other terminating errors to nonterminating errors by providing default values. Appendix D lists the locations of most of the terminating-error checkpoints.

SAMPLE RUN

Figure 10 gives a sample of the input for Project Invention when team 1 is making the first play. They have decided to award an incentive fee prime contract to contractor B to start on Jan. 1, 1970. Their target completion date for the project is Jan. 1, 1972. They are requesting all reports monthly at level 2. They also award the machinery design contract to contractor D as a fixed-fee contract although they do not want work to start until Jan. 1, 1971. No other teams are making inputs at this time, so the team 1 terminator is followed by a deck terminator 2. The action codes are based on the sample set shown in figure 8.

The following are some general restrictions on the input deck, based on the format of figure 10:

(1) All entries must be right justified within a particular field.

(2) The player (team) number and period number must appear on the first action card submitted for each play.

(3) The input for the teams must be in numerical order; i.e., team 1 input must precede team 2 and team 3 inputs. Each team's input is separated by a card with a 1 punched in column 1.

(4) If a team does not wish to take any actions for a period, but the history is to be updated 1 month, a card is required with the player number and period number followed by a 1 card.

(5) To update the project additional (one or more) periods, a type 3 terminator card is used instead of a type 2 card, followed by period, action, and type 1 cards for each team. Reports will be printed for each period.

(6) At the end of the input deck for all the players, there must be a card with a 2 punched in column 1.

(7) Actions are processed one at a time, beginning with the first action on the first card. Second and third actions on that card (if any) are then processed in order. Then the first action on the next card is processed. This sequence continues until all acquisitions for a period have been processed. All chronological references in the following rules (e.g., before or after) refer to the order in which actions are processed.

Terminator	First Action	Second Action	Third Action	Contract	System	Subsystem	Perturbation	Activity	Completion Schedule Date: Day	Mo	Yr	Assign Start Date: Day	Mo	Yr	XX	Dollars (in thousands)	Report	Player	Period	Predecessor Event	Successor Event	XX	Change	Remarks
1	2 3 4	5 6 7	8 9 10	11 12	13	14	15 16	17 18 19	20 21	22 23 24	25 26	27 28	29 30 31	32 33	34 35	36 37 38 39	40 41	42 43	44 45	46 47 48 49	50 51 52 53	54 55	56 57 58 59 60	61–80
																		1	1					
	2	9	6	1	0	0						31	DEC	69										
	7			1					1	JAN	72													
	10	15	16	1													1							
	11	15	16	1													2							
	12	15	16	1													3							
	13	15	16	1													4							
	14	15	16	1													5							
	4	8	6	2								31	DEC	70										
	10	15	16	2													1							
	11	15	16	2													2							
	12	15	16	2													3							
	13	15	16	2													4							
	14	15	16	2													5							
1																								
2																								

Figure 10.—Player's coding form.

(8) The first action number field must always be completed. More than one action may be taken on the same card by filling in the second and third action fields. If the second field is blank, the third field will be ignored. A perturbation card or a player number card, however, does not permit any action fields to be filled in.

(9) Any combination of actions may be taken on the same card if the required fields are compatible, except for the following: An action against a component and another action against an activity cannot be taken on the same card.

(10) A contract must be initiated before any other action referring to that contract is taken.

(11) When a contract is awarded, the contract start date and contract-type actions must follow on the same card. The first start date read by the program will also be used as the start date for that team's reports. No earlier start dates will be recognized for other contracts.

(12) Report frequencies and levels may not be set for reports that have not been requested, but reports may be requested before the contract is awarded. However, the costs of the reports will not be properly charged to the contract in this case.

(13) When a report is requested, the report frequency and level must be on one card. The request-report action must be in the first action field.

The referee should be provided with a list of actions indicating mandatory and optional entries for each action and a suitable coding form such as that of figure 10 for inputs not in his supply of prepunched cards. By limiting the number of potential contractors and acceptable contract types most combinations can be prepunched. It is desirable also to have prepunched cards for overtime, double shift, and triple shift for each activity as well as "cancel overtime," etc. The latter is needed if the team changes from overtime to double shift. These cards may be used by the harried referee to make time and cost changes regardless of the reason (if, for example, the team had a proper objection to the contractor's reprograming, it could be programed back somewhat by a "cancel overtime" card). Action NCODE 17, 20, or 21 may alternately be used, but the referee must enter the number of weeks change each time.

Referees conducting the OOO project have used three card cases as follows:

Box 1: Team-period header cards, contract award and report-request cards, perturbation request cards, and team terminator cards; total about one-third of a box

Box 2: Speedup cards: overtime, double shift, triple shift, weekend; one of each for each event pair (including perturbations) in the project; total nearly a full box

Box 3: Slowdown cards: cancel cards for each of box 2

REPORTS

Reports are selected for each contractor by setting values in the array NREP by NCODE 4, 5, or 6 action combinations. NREP provides space for the five report types from each of 15 contractors. The sample reports in appendix E resulted from the action inputs of figure 10.

Before producing the individual contractor's reports, however, the program always will issue the special referee reports and a project summary status report. These reports—not shown in the appendix—are as follows:

(1) Listing of input cards (less remarks) with error messages

(2) Summary history by months (estimated completion date and cost, etc.)

(3) Perturbation status report

(4) Listing of future activities that have been reprogramed more than 1 week from original value

(5) Overall probabilities for each subsystem

(6) Project summary status—a one-line project total of the amount of work done to date, overrun or underrun based on original value, estimated final completion date and cost, and unobligated funds available to project manager

Only the last report is given to the players.

The program scans the NREP array, computing and outputting each requested report in turn. Before calling the report subroutines, a check is made to see whether the contract is active. All of the type 1 reports will be produced, then the type 2 reports, etc. Because only one entry is possible for each report per contractor, it is not possible to have the report produced at a different level quarterly than monthly. The team should be informed that "since you are paying for a system capable of generating the details, you should use it each month."

The reports will be described in some detail so that they can be adapted to other forms. The subroutines are in FORTRAN.

Dates are carried in the arrays as tenths of weeks from the start of the project. The project start day, carried as tenths of weeks from January 1, and the project start year (last two digits) are stored in COMMON file. Each team may have a different value. Subroutine CLNDR converts the stored dates to day-month-year printer format.

Costs are carried in units of hundreds of dollars before overhead in the arrays. These are totaled for the particular sort requested, multiplied by the overhead for that contract, and truncated to print in thousands of dollars. Because of the truncation, the sum of individual lines as printed on a report may be less than the value printed as the total.

Report Level

Level 1 reports give information only for the contract and system. Level 2 reports give information for the contract, system, and system line items; i.e., engineering labor, material, technical labor, and subcontracts. Level 3 reports give information for the contract, system, system line items, and for the subsystem with subsystem line items.

The NREP array is set by the combination of NCODE (4), (5), and (6) actions as follows:

(1) 0 for not requested, or

(2) 1, 2, or 3 for level 1, 2, or 3 monthly reports, or

(3) -1, -2, or -3 for level 1, 2, or 3 reports quarterly—this suppresses printout of reports for the first 2 months of the quarter and produces the regular monthly report in the third month

Standard Types of Reports

Type 1—NASA Contractor Financial Management Report

This simulated report provides the maximum financial information available in the program and includes costs to date and projected spending to completion. Although the detailed projected spending plan is only prepared quarterly in real life, the program produces this each month. The contents are as follows

Item	*Description*
Player	Identification number of the participant asking for the report
Date	Month simulated by the data
Level	Report level (contract, system, or subsystem) of data to be generated
Description	The name of the contract for which the report is being generated from NCOMP (4, X) through NCOMP (9, X), where X is chosen such that NCOMP (2, X) and (3, X) = 0
Type	(Blank variable)
Value	Negotiated cost of this contract including reports—NCOMP (11) times NTRACT (5)
Ceiling	Maximum amount that can be paid to the contractor prior to completion of contract (currently blank)
Invoice amounts	70 percent of the to-date costs represents billing from the contractor = NTRACT (10, N)
Total payments	70 percent of the invoice amount = NTRACT (11, N)
Report date	End of the period simulated
Item	Variable according to level of the report; essentially, contract name, system name, subsystem name, or cost item breakout

The following are obtained by summation of the appropriate NACTIV (46) through (50):

Item	*Description*
Pd. costs	Proportion of the total cost of an activity that has been accrued during this simulated period

To date	Costs accrued from the time the contract is awarded to the end of the report date
Current qtr	The 3 months that make up the quarter in which the report falls
Mo. (1)	The proportion of the total cost of the activities that are ongoing in the first month of the quarter–if the first month of the quarter is the period, the costs will show only in the Pd. costs column and this column will contain zeros
Mo. (2)	The proportion of the total cost of the activities that are ongoing in the second month of the quarter–if the second month of the quarter is the period, the costs will show only in the Pd. costs column
Mo. (3)	The proportion of the total cost of the activities that are ongoing for the third month of the quarter
Qtr (2)	The proportion of the total cost of the activities that are ongoing for the 3 months following the quarter in which the report date falls
Qtr (3)	The proportion of the total cost of the activities that are ongoing for the 3 months following the second quarter
Qtr (4)	The proportion of the total cost of the activities that are ongoing for the 3 months following the third quarter
Qtr (5)	The proportion of the total cost of the activities that are ongoing for the 3 months following the fourth quarter
Bal. of F.Y.	Balance of fiscal year–the proportion of the total cost of the activities that are ongoing during the period starting the month after the fifth quarter and including all months through the next June
Next F.Y.	Next fiscal year–the proportion of the total cost of the activities that are ongoing from July through June of the next fiscal year
Bal. of contract	The proportion of the total cost of the activities that are ongoing from July of the fiscal year following "next F.Y." through the date the activities are to be completed
Total to compl	The sum of the costs of the 10 columns of data starting with month (1) through balance of contract
Est. final cost	The estimated cost of all the activities
Contract value	The negotiated cost of the component obtained from the base values
Est. compl date	The date when the component is expected to be completed

The sample report shown in appendix E is for contract 1, level 2. If requested for level 1, the first line of entries (labeled "Study CNTRCT") would have been the only output; if requested for level 3, additional detail lines would have been provided for each item of the work breakdown structure.

The report indicates an overrun of ($883 000 - $752 000) $131 000 or 17 percent of the planned costs. The only clue given by this report is that the completion date is about 6 months later than the date the team planned (not shown on this report).

A similar report was produced for the other contract but is not shown in the samples.

Type 2–NASA PERT Management Summary Report

This report is a PERT and companion cost report and can be selected at one of three levels of detail.

It is similar to the type 1 report, except that costs are not separated by months or quarters. The completion dates are converted to a graphical representation as shown in the sample and slack is given for each item.

An E is printed under the letter of the month in which the contract, system, or subsystem is expected to be completed. An L is printed under the letter of the month that is the latest allowable date for the contract, system, or subsystem. An S is printed under the letter of the month in which a schedule date for the network was set. The day of the month for each is printed in the columns headed E, L, and S.

For the sample shown, the latest allowable date for TRIAL VERS is read as 22 Jun 71 and the estimated date as 12 Jan 72. The left-hand letter of the calendar graphical representation will always be the current month. If a date is greater than 2 yr from the present date or is completed prior to the current month, it will be blank.

Because all of the items in this report sample show the same negative slack, it might be assumed that the start date for this contract was incorrectly chosen by the team and that the cost overruns are caused by free-slack charges. There is not sufficient detail in this report to verify this theory.

Note that reports are produced for any awarded contracts even though the actual work has not started.

Type 3–Schedule Analysis and Review Procedure Report

The schedule analysis and review procedure (SARP) milestone report prints those activities whose level NACTIV (9, J) value is nonzero and equal to or less than the report level requested. The base date for the activity is NACTIV (16, J), the preceding month's estimate is NACTIV (29, J), and the current estimate is NACTIV (42, J). If the activity has been completed, only the completion date is printed. Milestones are sorted in order of increasing date.

This report is an event-based report in real life. The identifier is the work breakdown number and the successor event number of the activity selected. For the report shown in appendix E, the first two items are for activity 11 (events 11 to 5) and activity 23 (events 8 to 5); activity 13, which also feeds into event 5 (events 4 to 5) was not chosen as a milestone. The game designer should avoid this potential confusion by selecting milestone events with only one incoming activity during design of the basic flowchart.

Type 4–NASA PERT Report

The NASA PERT printout is the listing of all the activities in the network belonging to all the contracts that have been awarded. It is produced in three sequences: successor sort, predecessor minor; expected date, predecessor minor; slack (paths of criticality), topological minor. Only the latter is shown in the sample in appendix E. The contents of all three are as follows:

Item	*Description*
Player	The identification number of the participant asking for the report
Date	The end of the month simulated by the data
Level	Always prints a 2
Beginning date of network	The initial project start date
First run	The date of the first network pass
Updated	The date of last input data
Pred event	The component and predecessor event number of the activity; NACTIV (4, J), (5, J), (6, J), and (2, J)
Succ event	The component and successor event number of the activity; NACTIV (4, J), (5, J), (6, J), and (2, J)
Activity description	The activity description; NACTIV (55, J) through (66, J)
Expected date	The date on which the activity is expected to be completed; NACTIV (42, J) CONVERTED
Allowed date	The latest allowable date at which the activity can be completed and meet the project schedule; NACTIV (43, J) CONVERTED
Activity complete	Yes/no depending on whether activity is completed; bit 13 of NACTIV (54, J)
Schld/act date	Scheduled date if activity is the terminal one in the component or network and a schedule date was set; otherwise, the date the activity was completed; blank if not completed or no schedule date set
Slack	Sign and total activity slack in tenths of weeks; NACTIV (44, J)
Time from beginning	The sum of the durations of all predecessor activities; the expected date in tenths of weeks; NACTIV (42, J) (Note: An asterisk following the time indicates that the player has assigned a specific start date to this activity.)

The sample shown verifies the previous judgment that the team chose an incorrect start date for contract 2. To change this date, a referee input containing only the start date action number (6) and a new date is used (the contract is not reawarded).

A comparison of the individual activity times listed in this report and those originally shown on the flowchart (fig. 4) or the PREEP input (fig. 6) confirms the fact that the contractor's present estimates have substantially increased over his original proposal. For example, activity from event 3 to event 4 (was 14.6, now 20.6 weeks), or activity leading from event 10 to 11 (was 12.0, now 13.3 weeks). The total increase in duration is about 20 weeks beyond an original duration of about 110 weeks. Later reprograming will not be as severe as that after the first play, as there will be smaller reliability changes and the RANDOM entry is greatest in January.

If the designer finds the project changes not realistic during the life of the game, the easiest method of correction is to apply probability changes to the whole contract for all contractors in the initial award contract actions. In the example from the OOO project (tables 1 and 2), generally negative trends were incorporated in these actions to increase the rate of deterioration instead of changing entries on most of the activity cards (type 10).

Type 5–Technical Narrative Report

This report prints those activities completed during the current month (bit 6 in NACTIV (54, J) = 1), those activities expected to be completed during the next month, and test activities that have failed (bit 8 in NACTIV (54, J)). The overall reliability average for the contract and the latest completion date for any activity in the contract are also found and printed.

The program assumes that contract 12 is a reliability study and prints a special report if NREP (5, 12) is called, which lists the reliability of each subsystem. To remove this type of report, the programer should delete statements following 900 in REPGEN.

Miscellaneous Reports

When a contract is canceled after work has started, by NCODE = (15), all reports on this contract will cease. However, a final type 1 report is issued showing the contract termination and amount of funds returned to the project. Termination costs are nominally 10 percent of the value of the work completed. If the contract is reawarded, the new contract will show charges again for the work completed; therefore the referee should avoid this situation.

Provision is made through use of an action calling NCODE = (16) for a planning run. The reports produced are a normal type 4 (PERT) report, an estimated-cost-to-complete report giving only the final cost for each contract, and an estimated-reliability report from contract 12. The game assumption for these reports is that they are the predictions of the project's support staff based on new management plans. The model obtains these results by backing up to the last-month values stored in the arrays, deactivating any perturbations and all other reports, adding the new management decisions specified by the team, and then performing a normal computation. The output magnetic tape storing the player's position after this run should not be used for further normal runs; thus all teams should make their planning runs at the same time to avoid extra computer job submissions.

Additional reports may be added by increasing the 5 dimension of NREP and adding new subroutine CALL's in REPGEN. The existing reports are designed to represent standard NASA reports. Headings may be easily changed. The basic data for the reports are stored in the COMMON arrays.

REDUCTION OF CORE REQUIREMENTS

The PREEP and MAIN programs must have the same dimensions in the storage arrays read from the player's tapes. The principal arrays are stored in unnamed COMMON. Considerable core storage may be saved if only a small network is being used, by redimensioning these in all subroutines. The detailed contents of each major array are listed in appendix B. These arrays are defined for the IBM 360 version as follows:

NACTIV (66, 999)	One row of 66 words for each of up to 999 event pairs in the network
NCOMP (48, 50)	One row of 48 words for each of up to 50 components (subsystems)
NTRACT (13, 15)	One row of 13 words for up to 15 contracts
NREP (15, 5)	Five types of reports for each of 15 contracts
NCPDAT (5, 30)	Five items of history for up to 30 plays per player
MONS (12)	Calendar constant
NDATE (12)	Calendar constant
NDAYS (12)	Calendar constant
NCOM (22)	Communications region (see below)

A named COMMON area LOCAL containing the following is used during the first two phases of play:

NACTN (16, 120) NCODE	Cost time-and-reliability parameters for up to 120 actions (the PREEP data file)
PERB (21, 99)	Perturbation changes for up to 99 perturbations
NDIST (99)	Distribution curve (fig. 2)
IN (22)	Player's card input being processed
NAA (3, 120)	120 actions taken against activities for this play
NAC (5, 120)	120 actions taken against contracts or subsystems for this play

To save searching the complete array when it is not filled with data, the number of items of most arrays is placed in NCOM for use as DO loop limits. Thus array dimensions may be redefined without extensive program modifications.

Array NCOM contains–

(1) Actual number of activities in NACTIV

(2) Actual number of perturbations in PERB

(3) Actual number of components in NCOMP

(4) Actual number of contracts in NTRACT

(5) Actual number of actions in NACT

(6) Player number for this file

(7) Number of distribution entries in NDIST

(8) Period number of last play

(9) Error flag

(10) Base week of contract 1 origin (in tenths of weeks)

(11) Base year of contract 1 origin

(12) (Not used)

(13) (Not used)

(14) Random number

(15) Maximum expected completion time (weeks and tenths of weeks from start date)

(16) (Not used)

(17) (Not used)

(18) Total number of player files

(19) Budget balance bin (in hundreds of dollars)

(20) Current year

(21) Current month

(22) Input-deck terminator flag

Similarly, KA and KC in LOCAL contain the actual number of actions stored in NAA and NAC for each play, and NPLRS is the player number being processed, which should equal NCOM (6).

Goddard Space Flight Center
National Aeronautics and Space Administration
Greenbelt, Maryland, July 20, 1972
039-06-01-02-51

Appendix A

ACTION SUBROUTINES

The following is a list by number of the NCODE's currently implemented in the main program. This list shows where data are required in establishing the action with the PREEP program by use of ACTION file cards and on other data cards at the time of play. Figure 8 shows coding for the PREEP type 40 cards and figure 10 shows coding for play. Probability data are optional for all NCODE's unless noted. The OOO project referee manual (obtainable from COSMIC) provides additional examples of usage.

(1) NCODE = (1) calls subroutine ACTNA in the ACTION subroutine. It initializes any contract by setting the existence indicator bit in all activities (except perturbations) for the specified contract. It also picks up the overhead cost multiplier and determines the total contract value, subtracting this value from the project budget.

PREEP required data—the contract number and the overhead value. A separate action number is required for each contract. (There may be more than one for each contract, as in table 1, showing representative values of changing costs and probabilities to represent the differences among contractors for the OOO project.)

Play required data—contract number (used as an error check) and an NCODE = (2) action on the same card.

(2) ACTNB sets start dates for contracts or subsystems. The first start date read by the playing program is also used to set the date on the output reports. Work will start on the day following the date on the play card. Thus for the contract to be dated 1 Jan 70, the entry would be 31 Dec 69. Contracts must start on the first of a month.

PREEP required data—none

Play required data—contract number and start date

(3) ACTNS adds funds to a contract (or system) without deducting from the project budget. This simulates additional funds appropriated for the specific contract.

PREEP required data—none

Play required data—contract number and dollar value

(4) ACTND sets the report request for the type of report. It also sums report cost multipliers $U(c)$ in the 40 file. In the play program, this NCODE also picks up costs from actions with

NCODE = (5) and NCODE = (6), which must be on the same player's card. These costs are added to the contract value and subtracted from the project budget. A separate action number is required for each report number.

PREEP required data—report number in *C/R/L* entry, cost multiplier in $U(c)$

Play required data—contract number and report number

(5) ACTNE sets the level or detail of the report (level 1, 2, or 3 for cost reports, level 1 or 2 for time reports).

PREEP required data—level number in *C/R/L* entry, cost multiplier

Play required data—contract number, report number

(6) ACTNF sets the frequency (monthly or quarterly) of the report.

PREEP required data—frequency number: 0 for monthly and 1 for quarterly in *C/R/L* entry; cost multiplier

Play required data—contract number, report number

(7) ACTNG cancels a previously requested report.

PREEP required data—The report number must be in the *C/R/L* column. It is suggested that the negative value for $U(c)$ representing the average value of a report system, as well as negative values for P or S, be included on these 40 cards to compensate for the effect of the contractor canceling these reports.

Play required data—contract number and report number.

(8) ACTNH sets the final schedule completion date for the project if the player's card specifies contract 1, otherwise it sets a due date for a contract, system, or subsystem (not used in slack computation but merely printed on reports).

PREEP required data—none

Play required data—contract number and completion date

(9) ACTNI determines from random numbers and the progress of work to date, what proportion of a fund request will be approved. It then goes to subroutine ACTNS. (See NCODE = (3).) No data are required in the 40 file; however, some probability changes may be useful.

PREEP required data—none

Play required data—contract number, dollar value

(10) ACTNJ subtracts funds from the project budget and then goes to subroutine ACTNS to add them to a specified contract. It checks that a positive sign is associated with the funds specified on the player's input card.

PREEP required data—none

Play required data—contract number, dollar value

(11) Same as NCODE = (10), except this routine removes the funds from the contract, returns them to the project budget, and checks for a minus sign on the player's input card.

PREEP required data—none

Play required data—contract number, dollar value

(12) ACTNT removes a system or subsystem from further consideration in the project by turning off the existence indicators in all of its activities. It then goes to subroutine ACTNU to correct the contract value and return the funds to the project budget. If submitted a second time against the same system or subsystem, it will restore it for consideration.

Notes: This routine could be used for company C in the example of table 1 to delete Government-furnished equipment from the network. Care must be taken that deletion of these activities does not leave a gap or discontinuity in the flowchart. Canceling after active work has begun and then restarting may produce inconsistent numerical results.

PREEP required data—none

Play required data—system or subsystem number

(13) ACTNC sets the start date for an activity.

Notes: An asterisk will be printed at the end of the line for this activity in all PERT reports to indicate the time discontinuity. This action cannot be canceled, but revised dates may be submitted.

PREEP required data—none

Play required data—either an activity number or a predecessor-successor event pair and the date (day, month, year)

(14) This NCODE currently has no specific effect in the program. It may be used for actions to apply fixed probability, time, or cost changes as set by the 40 file data to a system or to an activity. (For an example, see line 40-18 in fig. 8 for simulating overtime work.)

PREEP required data—none

Play required data—either contract, system or subsystem, activity, or event pair number

(15) ACTNN cancels a whole contract. It is similar to NCODE = (12) except that a final financial report is produced on closeout of the contract, and the contract cannot be reinstated without confusion in the amount of work that has been done. It is recommended that the contract not be reinstated if canceled by this action.

PREEP required data—none

Play required data—contract number

(16) This NCODE is used to produce a planning run. Time does not update 1 month, and no perturbations will activate. Only a summary cost report, the PERT reports, and the reliability report are produced.

PREEP required data—none

Play required data—none

(17) ACTNR is used to make a positive or negative change in the duration of a single activity with a corresponding change in costs.

PREEP required data—none (probability changes may be useful)

Play required data—activity or event pair, time change in whole weeks

(18) Same as NCODE (14).

(19) Same as NCODE (14).

(20) ACTNR; same as NCODE = (17), except no associated cost change.

(21) ACTNR; same as NCODE = (17), except the cost change is of opposite sign to the time change.

(22) ACTNX makes a change in the probabilities (for a single activity or for systems and subsystems) in terms of the percentage change specified on the player's input card.

PREEP required data—there must be a nonzero entry in $P(t)$, $P(r)$, or $P(c)$ for change to be effective. The value of the entry in the PREEP file has no numerical effect on the probabilities.

Play required data—activity, event pair, or system or subsystem number and a signed value for percent of change.

(23) This NCODE operates within subroutine ACTION to establish or change the total funds available to the project.

PREEP required data—none

Play required data—dollar value of change (in thousands)

(24) This NCODE calls subroutine ACTNY to change the cost of a particular activity or of a subsystem without changing the duration of work.

PREEP required data—none

Play required data—contract, system or subsystem, event pair, or activity number and the dollar change (in thousands)

(25) This NCODE is used for the OOO project to initiate immediately perturbation 11 and to apply probability changes at the same time through the use of a single action number.

PREEP required data—probability and cost changes (optional)

Play required data—none

(26) Same as NCODE (25) except perturbation = 20.

(27) Same as NCODE (25) except perturbation = 34.

(28) Same as NCODE (25) except perturbation = 35.

Appendix B

IBM 360 FILE CONTENTS

NACTIV (66, 999) ACTIVITY FILE

(1, J)	Predecessor event number
(2, J)	Successor event number
(3, J)	Activity number
(4, J)	Contract number
(5, J)	System number
(6, J)	Subsystem number
(7, J)	Perturbation number
(8, J)	Time deviation (base)
(9, J)	Activity level indicator
(10, J)	Reliability character
(11, J)	Cost deviation (base)
(12, J)	Time probability (base)
(13, J)	Cost probability (base)
(14, J)	Reliability probability (base)
(15, J)	Duration (base)
(16, J)	Expected completion (base)
(17, J)	Latest completion (base)
(18, J)	Total slack (base)
(19, J)	Free slack (base)
(20, J)	Activity total cost (base)
(21, J)	Activity engineering cost (base)
(22, J)	Activity material cost (base)
(23, J)	Activity technical labor cost (base)
(24, J)	Activity subcontract cost (base)
(25, J)	Time probability (last month)
(26, J)	Cost probability (last month)
(27, J)	Reliability probability (last month)
(28, J)	Duration (last month)
(29, J)	Expected completion (last month)
(30, J)	Latest completion (last month)
(31, J)	Total slack (last month)

(32, J)	Free slack (last month)
(33, J)	Activity total cost (last month)
(34, J)	Activity engineering cost (last month)
(35, J)	Activity material cost (last month)
(36, J)	Activity technical labor cost (last month)
(37, J)	Activity subcontract cost (last month)
(38, J)	Time probability (this month)
(39, J)	Cost probability (this month)
(40, J)	Reliability probability (this month)
(41, J)	Duration (this month)
(42, J)	Expected completion (this month)
(43, J)	Latest completion (this month)
(44, J)	Total slack (this month)
(45, J)	Free slack (this month)
(46, J)	Activity total cost (this month)
(47, J)	Activity engineering cost (this month)
(48, J)	Activity material cost (this month)
(49, J)	Activity technical labor cost (this month)
(50, J)	Activity subcontract cost (this month)
(51, J)	Work rate multiplier
(52, J)	Start day in weeks (if set by ACTNC)
(53, J)	(Used by ACTION)
(54, J)	Packed word (used by PACK and UNPAK)
(55, J)	Description Hollerith (1st word, 2 characters)
(56, J)	Description Hollerith (2d word, 2 characters)
(57, J)	Description Hollerith (3d word, 2 characters)
(58, J)	Description Hollerith (4th word, 2 characters)
(59, J)	Description Hollerith (5th word, 2 characters)
(60, J)	Description Hollerith (6th word, 2 characters)
(61, J)	Description Hollerith (7th word, 2 characters)
(62, J)	Description Hollerith (8th word, 2 characters)
(63, J)	Description Hollerith (9th word, 2 characters)
(64, J)	Description Hollerith (10th word, 2 characters)
(65, J)	Description Hollerith (11th word, 2 characters)
(66, J)	Description Hollerith (12th word, 2 characters)

PACKED-WORD NACTIV (54, J)

Bit number	*Function*
1	(Not used)
2	Remove overtime indicator
3	Freeze indicator (voluntary)
4	Freeze indicator (automatic)

5	Component terminal activity indicator
6	Just completed indicator
7	Start date indicator (set by ACTNC)
8	Failure indicator
9	Test activity indicator
10	Terminal activity indicator
11	Start indicator
12	Begin indicator
13	Completion indicator
14	Work rate indicator
15	Existence indicator
16	Initial activity indicator

NCOMP (48, 50) COMPONENT FILE

(1, J)	Contract number
(2, J)	System number
(3, J)	Subsystem number
(4, J)	Description Hollerith (1st word, 2 characters)
(5, J)	Description Hollerith (2d word, 2 characters)
(6, J)	Description Hollerith (3d word, 2 characters)
(7, J)	Description Hollerith (4th word, 2 characters)
(8, J)	Description Hollerith (5th word, 2 characters)
(9, J)	Description Hollerith (6th word, 2 characters)
(10, J)	Scheduled completion date
(11, J)	Component total cost to completion (base)
(12, J)	Component engineering cost (base)
(13, J)	Component material cost (base)
(14, J)	Component technical labor cost (base)
(15, J)	Component subcontract cost (base)
(16, J)	Time probability (base)
(17, J)	Cost probability (base)
(18, J)	Reliability probability (base)
(19, J)	Expected completion (base)
(20, J)	Component total cost to completion (last month)
(21, J)	Component engineering cost (last month)
(22, J)	Component material cost (last month)
(23, J)	Component technical labor cost (last month)
(24, J)	Component subcontract cost (last month)
(25, J)	Component time probability (last month)
(26, J)	Component cost probability (last month)
(27, J)	Component reliability probability (last month)
(28, J)	Expected completion (last month)
(29, J)	Component total cost to completion (this month)

(30, J)	Component engineering cost (this month)
(31, J)	Component material cost (this month)
(32, J)	Component technical labor cost (this month)
(33, J)	Component subcontract cost (this month)
(34, J)	Component time probability (this month)
(35, J)	Component cost probability (this month)
(36, J)	Component reliability probability (this month)
(37, J)	Expected completion (this month)
(38, J)	Total cost expended (last month)
(39, J)	Engineering cost expended (last month)
(40, J)	Material cost expended (last month)
(41, J)	Technical labor cost expended (last month)
(42, J)	Subcontract cost expended (last month)
(43, J)	Total cost expended (this month)
(44, J)	Engineering cost expended (this month)
(45, J)	Material cost expended (this month)
(46, J)	Technical labor cost expended (this month)
(47, J)	Subcontract cost expended (this month)
(48, J)	Start date (if set by ACTN)

NTRACT (13, 15) CONTRACT FILE

(1, J)	Contract number
(2, J)	Start date
(3, J)	Scheduled completion date
(4, J)	Authorized cost (base)
(5, J)	Overhead multiplier
(6, J)	Freeze period
(7, J)	Freeze activity hit
(8, J)	Estimated overrun
(9, J)	Authorized funding
(10, J)	Invoiced amount (this month)
(11, J)	Payments (this month)
(12, J)	Unfreeze date
(13, J)	Contract initiated indicator

ARRAY NACTN (16, 120)

(1, J)	Action number
(2, J)	Time-probability multiplier $P(t)$
(3, J)	Cost-probability multiplier $P(c)$
(4, J)	Reliability-probability multiplier $P(r)$
(5, J)	Not used (formerly $Q(t)$)
(6, J)	Not used (formerly $Q(c)$)

(7, J)	Not used (formerly $Q(r)$)
(8, J)	Time-probability additive $S(t)$
(9, J)	Cost-probability additive $S(c)$
(10, J)	Reliability-probability additive $S(r)$
(11, J)	NCODE for subroutine call
(12, J)	Percent change in time $U(t)$
(13, J)	Percent change in cost $U(c)$
(14, J)	Cost dollars change $G(c)$
(15, J)	Overhead multiplier
(16, J)	$C/R/L$

Appendix C

PREEP TERMINATING ERRORS

Error number	*Cause*
500	First card of data deck does not contain a 75 in columns 1 and 2.
501	Type 10 card detected before type 20 file.
502	No input-deck terminator for perturbation data.
503	Perturbation number in perturbation card is out of sequence; must be in ascending sequence.
504	More than 50 perturbations.
505	Delete or add activities missing after perturbation card.
506	Delete or add activities that do not have the same perturbation number as the perturbation.
507	Perturbation file cards are not in order—cards 30, 31, 32.
508	Activity number is out of sequence; activities must be in ascending sequence.
509	More than 523 activities in activity file.
510	Sum of breakdown costs does not equal total cost.
511	No input-deck terminator for activity data.
512	More than 60 components.
513	Month card of calendar deck does not contain a 50 in columns 1 and 2.
515	No input-deck terminator at end of component file.
516	Action number is missing from action file.
517	More than 100 actions.
518	No input-deck terminator at end of action deck.
521	No input-deck terminator at end of distribution data.
522	There are not 99 values for the distribution file.
523	Data card for one of 5 files was inputted after the input-deck terminator for that file has been inputted.
524	First card of distribution file does not contain a 60 in columns 1 and 2.
525	No input-deck terminator at end of calendar deck.
526	Date card of calendar file does not contain a 50 in columns 1 and 2.

Appendix D

MAIN GREMEX TERMINATING ERRORS

Error number	Identifier	Cause of error	Probable calling subroutine
1	1	2d or 3d action on "award contract" not a set start date.	ACTION
	([a])	Input contract number not equal to action contract number.	ACTNA
2	1	Incorrect action number.	ACTION
	4	Input report number not equal to action report number.	ACTND
	5	Input contract number not found in file data.	ACTNE
	7	Input contract number not found in file data.	ACTNG
	9	Input contract number not found in file data.	ACTNI
	41	Input contract number not found in file data.	ACTND
	51	Report has not been requested.	ACTNE
	81	Day, month, or year input is zero or blank.	ACTNH
	82	Project start date has not been set.	ACTNH
	83	No match found for input component data.	ACTNH
	91	Project start date has not been set.	ACTNI
	102	Input contract number not found in file.	ACTNJ
	140	Input contract number not found in file.	ACTNN
	103	Sign on money card does not agree with action number requirement.	ACTNJ
3	1	Date punch is zero.	ACTNB
	2	Month punch is zero.	ACTNB
	3	Month punch is blank.	ACTNB
	4	Year punch is zero.	ACTNB
	5	Month punch is invalid.	ACTNB
	6	Number of days punched is too large.	ACTNB
4	([a])	Contract for this component not active (in activity file).	ACTNB
5	([a])	No match found for component number in activity file.	ACTNB
	(Third action number)	2d or 3d action number wrong for request report card.	ACTION
6	([a])	No match found for component number in component file.	ACTNB
7	([a])	No match found for contract number in contract file.	ACTNB
10	(PERTB number)	This number not in PREEP file.	SETPTB
11	(Activity number)	Activity to be deleted by PERTB not in activity file.	FORWAR
12	(Activity number)	Activity to be added by PERTB not in activity file.	FORWAR
13	([a])	Component number not in component file.	CSRCH

Error number	Identifier	Cause of error	Probable calling subroutine
14	(Activity number)	Critical activity number for perturbation not in activity file.	FORWAR
20	([a])	Input contract number not in files.	ACTNA
25	([a])	This number from activity file not found in contract file.	FORWAR
31	1	Start date must be set for contract first.	ACTNB
90	0	Month characters not first 3 letters of a month.	DTCONV
102	(Player number)	Project not yet initialized.	FORWAR
103	(Predecessor number)	Predecessor-successor pair not in activity file.	ACTION
150	(Action number)	NCODE in action file is incorrect.	ACTION
237	(Period number not correct)	Player's card has wrong period number.	ACTION

[a]Identifier prints the contract number (or first part of component number) from column 5 of the input card.

Appendix E

TYPES OF STANDARD REPORTS IN GREMEX

PLAYER	1
DATE	31 JAN 70
LEVEL	2
PAGE	1

(FIGURES IN THOUSANDS OF DOLLARS)

DESCRIPTION	TYPE	VALUE	CEILING	INVOICE AMOUNTS	TOTAL PAYMENTS	REPORT DATE
STUDY CNTRCT		752		33	23	31 JAN 70

ITEM	PD. COSTS	TO DATE	CURRENT QTR MO. (1)	MO. (2)	MO. (3)	QTR (2)	QTR (3)	QTR (4)	QTR (5)	BAL. OF F.Y.	NEXT F.Y.	BAL. OF CON-TRACT	TOTAL TO COMPL.	EST. FINAL COST	CON-TRACT VALUE	EST. COMPL DATE
STUDY CNTRCT	48	48	0	69	94	223	110	92	160	85	2	0	835	883	752	25 JUL 72
FIRST DESIGN	48	48	0	69	94	208	29	35	0	15	0	0	450	498	445	05 JUL 71
ENGR LABOR	15	15	0	10	10	62	12	25	0	15	0	0	134	149	123	
MATERIALS	17	17	0	29	15	19	4	0	0	0	0	C	67	84	83	
TECH LABOR	17	17	0	17	27	56	8	12	0	0	0	0	120	137	131	
SUBCONTRACT	0	0	0	10	42	73	0	0	0	0	0	0	125	125	116	
FINAL DESIGN	0	0	0	0	0	15	81	56	160	71	2	0	385	385	307	25 JUL 72
ENGR LABOR	0	0	0	0	0	15	81	54	10	0	2	0	162	162	125	
MATERIALS	0	0	0	0	0	0	0	0	137	71	0	0	208	208	183	
TECH LABOR	0	0	0	0	0	0	0	2	12	0	0	0	14	14	10	
SUBCONTRACT	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure E-1.—Type 1, NASA contractor financial management report.

PLAYER 1
DATE 31 JAN 70
LEVEL 2
PAGE 1

CONTRACT MACHINERY CO — CONTRACTOR VARIOUS — REPORT DATES 31DEC 69 TO 31JAN 70

E - EARLIEST COMPL DATE
S - SCHED COMPL DATE
L - LATEST COMPL DATE
(SEE NOTE BELOW)

	WORK PERFORMED TO DATE			TOTALS AT COMPLETION							
ITEM	VALUE	ACTUAL COST	(OVERRUN) UNDERRUN	PLANNED COST	LATEST REVISED EST	PROJECTED (OVERRUN) UNDERRUN	JFMAMJJASONDJFMAMJJASONDJFMAMJ	DAY E	DAY L	DAY S	SLACK (WEEKS)
MACHINERY CO	0	0	0	241	244	(3)	L E	26	04		-29.2
TRIAL VERS	0	0	0	118	122	(4)	L E	12	22		-29.2
ENGR LABOR	0	0	0	57	59	(2)					
MATERIALS	0	0	0	10	10	0					
TECH LABOR	0	0	0	45	46	(1)					
SUBCONTRACT	0	0	0	5	5	0					
PRODUCTION	0	0	0	118	122	(4)	L E	26	04		-29.2
ENGR LABOR	0	0	0	1	1	0					
MATERIALS	0	0	0	0	0	0					
TECH LABOR	0	0	0	0	0	0					
SUBCONTRACT	0	0	0	126	120	6					

NOTE - - EXPLANATION OF SYMBOLS OTHER THAN E, L, S.

B	E AND L TOGETHER
G	E AND S TOGETHER
N	L AND S TOGETHER
RECORD MARK	E, L, S TOGETHER

Figure E-2.—Type 2, NASA PERT management summary report.

PLAYER 1
DATE 31 JAN 70
LEVEL 2
PAGE 1

PROJECT		MILESTONE NO.	MILESTONE DESCRIPTION	ACTIVITY COMPLETE	COMPL DATE	ESTIMATED COMPLETION DATES BASE	PREV	CURR
STUDY CNTRCT	111	5	ASSEMBLE PARTS	NO		C5 APR 70	05 APR 70	12 MAY 70
	111	5	ASSEMBLE SPECIAL PARTS	NO		09 JUN 70	09 JUN 70	20 JUN 70
	111	4	MACHINE BASIC FRAMES	NO		25 JUN 70	25 JUN 70	20 AUG 70
	111	70	DURABILITY TESTS	NO		15 OCT 70	15 OCT 70	26 DEC 70
	123	3000	PRODUCT.FINAL DECIS.	NO		03 SEP 70	03 SEP 70	20 JUL 71
	121	3000	APPROVE MACHINE FUNCTION	NO		10 APR 71	10 APR 71	30 JAN 72
	123	3001	MGMNT APPROVAL RCVD.	NO		11 SEP 71	11 SEP 71	25 JUL 72

Figure E-3.—Type 3, schedule analysis and review procedure (SARP) report.

PLAYER 1
DATE 31 JAN 70
LEVEL 2
PAGE 1

BY PATHS OF CRITICALITY
NETWORK PROJECT COO

BEGINNING DATE OF NETWORK IS 1 JAN 70
FIRST RUN 1 JAN 70 UPDATED 31 JAN 70

PREC EVENT	SUCC EVENT	ACTIVITY DESCRIPTION	ACTIVITY TIME	EXPECTED DATE	AL_OWED DATE	ACTIVITY COMPLETE	SCHLD/ACT DATE	SLACK	TIME FROM BEGINNING
210 50	210 51	SURVEY PRODUCTION METHOD	12.2	27 MAR 71	03 SEP 70	NO		-29.2	64.3
210 51	210 56	MODIFY MACHINE DESIGNS	13.6	30 JUN 71	07 DEC 70	NO		-29.2	77.9
210 56	210 57	FABRICATE MACHINE CHANGE	20.5	20 NOV 71	30 APR 71	NO		-29.2	98.4
210 57	210 60	TRIAL RUNS COMPL.	7.6	12 JAN 72	22 JUN 71	NO		-29.2	106.0
220 80	2202060	PRODUCTION RUN	19.8	31 MAY 72	08 NOV 71	NO		-29.2	125.8
2202080	2203000	EVALUATE PRODUCTION RUN	3.8	26 JUN 72	04 DEC 71	NO		-29.2	129.6
1233000	1233001	MGMNT APPROVAL RCVD.	4.1	25 JUL 72	02 JAN 72	NO	02 JAN 72	-29.2	133.7
121 80	1213000	APPROVE MACHINE FUNCTION	2.6	30 JAN 72	04 DEC 71	NO		-8.2	108.6
111 1	111 2	AWARD PRIME CONTRACT	0.0		06 JAN 70	YES	01 JAN 70	0.7	0.0
111 2	111 3	SCHEDULE MODEL SHOP	12.5	29 MAR 70	03 APR 70	NO		0.7	12.5
111 3	111 4	MACHINE BASIC FRAMES	20.6	20 AUG 70	25 AUG 70	NO		0.7	33.1
111 4	111 5	DIP + PLATE COMPLETE	12.2	14 NOV 70	18 NOV 70	NO		0.7	45.3
111 5	111 70	DURABILITY TESTS	6.0	26 DEC 70	30 DEC 70	NO		0.7	51.3
121 70	1212000	REVISE FINAL DESIGN	4.9	29 JAN 71	03 FEB 71	NO		0.7	56.2
1212000	1212001	ORDER PRODUCTION SUPPLIE	2.0	12 FEB 71	17 FEB 71	NO		0.7	58.2
1212001	121 57	MATERIALS PROCURED	10.3	25 APR 71	30 APR 71	NO		0.7	68.5
111 2	111 9	PREPARE SHOP DRAWINGS	6.4	14 FEB 70	01 APR 70	NO		6.5	6.4
111 9	111 3	REPRODUCE DRAWINGS	0.3	17 FEB 70	03 APR 70	NO		6.5	6.7
210 51	210 110	EVALUATE SURVEY REPORT	4.2	25 APR 71	09 SEP 71	NO		19.6	68.5
113 110	113 120	MARKET SURVEY (PRELIM)	10.2	05 JUL 71	19 NOV 71	NO		19.6	78.7
123 120	1233000	PRODUCT.FINAL DECIS.	2.1	20 JUL 71	04 DEC 71	NO		19.6	80.8
111 9	111 10	PREPARE SPECIAL SPECS	0.8	20 FEB 70	21 JUL 70	NO		21.6	7.2
111 10	111 11	SPECIAL MATERIALS RCVD.	13.3	24 MAY 70	22 OCT 70	NO		21.6	20.5
111 11	111 5	ASSEMBLE SPECIAL PARTS	3.9	20 JUN 70	18 NOV 70	NO		21.6	24.4
111 2	111 6	PREPARE STOCK SPECS	2.5		27 JUL 70	YES	18 JAN 70	27.2	2.5
111 6	111 7	COMMECIAL PARTS RCVD.	8.0	15 MAR 70	21 SEP 70	NO		27.2	10.5
111 7	111 8	INCOMING INSPECTION	4.0	12 APR 70	19 OCT 70	NO		27.2	14.5
111 8	111 5	ASSEMBLE PARTS	4.3	12 MAY 70	18 NOV 70	NO		27.2	18.8
112 9	1121000	REDRAW IN LEGAL FORMAT	6.2	30 MAR 70	05 APR 71	NO		53.1	12.6
1121000	1121001	INITIAL PATENT SEARCH	8.2	26 MAY 70	02 JUN 71	NO		53.1	20.8
1221001	1221002	SECOND PATENT STUDY	4.1	24 JUN 70	30 JUN 71	NO		53.1	24.9
1221002	1221003	LEGAL REPORT RCVD.	2.0	08 JUL 70	14 JUL 71	NO		53.1	26.9
1221003	1223000	FINAL LEGAL APPROVE	20.4	28 NOV 70	04 DEC 71	NO		53.1	47.3
113 2	113 100	HIRE ADDITIONAL STAFF	8.1	26 FEB 70	26 MAY 71	NO		64.8	8.1
113 100	113 110	PRELIMINARY SALES STUDY	15.2	13 JUN 70	09 SEP 71	NO		64.8	23.3

Figure E-4.—Type 4, NASA PERT report.

```
                          PLAYER          1
                          DATE      31 JAN 70
                          LEVEL           2
                          PAGE            1

              COMPONENT
            IDENTIFICATION        DESCRIPTION
               1 0 0                     STUDY CNTRCT



                ACTIVITIES COMPLETED LAST MONTH
PRED EVENT   SUCC EVENT
    1             2      AWARD PRIME CONTRACT
    2             6      PREPARE STOCK SPECS



         ACTIVITIES EXPECTED TO BE COMPLETED NEXT MONTH
PRED EVENT   SUCC EVENT
    2           100      HIRE ADDITIONAL STAFF
    9             3      REPRODUCE DRAWINGS



                         TEST FAILURES
PRED EVENT   SUCC EVENT



  EXPECTED TECHNICAL PERFORMANCE            0.65
  EXPECTED COMPLETION DATE             25 JUL 72
```

Figure E-5.—Type 5, technical narrative report.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
WASHINGTON, D.C. 20546

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE $300

SPECIAL FOURTH-CLASS RATE
BOOK

POSTAGE AND FEES PAID
NATIONAL AERONAUTICS AND
SPACE ADMINISTRATION
451

U.S.MAIL

POSTMASTER : If Undeliverable (Section 158 Postal Manual) Do Not Return

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons. Also includes conference proceedings with either limited or unlimited distribution.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include final reports of major projects, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION OFFICE

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546